



ESTG



INSTITUTO POLITÉCNICO
DE VIANA DO CASTELO

Plataforma para Gestão de Oficinas

Mestrado em Engenharia de Software

Dissertação efetuada sob orientação de
Professor Doutor Pedro Miguel Faria
Professor Doutor Pedro Miguel Moreira

João António Guimarães Ribeiro

Plataforma para Gestão de Oficinas
João António Guimarães Ribeiro

2019

Escola Superior de Tecnológica e Gestão



INSTITUTO POLITÉCNICO
DE VIANA DO CASTELO

João António Guimarães Ribeiro
Plataforma de Gestão para Oficinas

Mestrado em Engenharia de Software

Dissertação efetuada sob orientação de
Professor Doutor Pedro Miguel Faria
Professor Doutor Pedro Miguel Moreira

fevereiro de 2019

AGRADECIMENTOS

Agradeço aos meus orientadores, Professor Doutor Pedro Miguel Faria e Professor Doutor Pedro Miguel Moreira, por toda a ajuda e disponibilidade prestada.

Agradeço à minha família, em especial aos meus pais e irmã pela ajuda e apoio durante todo o meu percurso académico.

Agradeço aos meus amigos, em especial ao Pedro Barbosa por todo o apoio durante a realização desta dissertação.

RESUMO

Atualmente o mercado de smartphones tem vindo a crescer de um modo muito significativo, proporcionando ao cidadão possuir um destes dispositivos.

De modo a desenvolver o projeto descrito nesta dissertação, foram contactadas diversas oficinas do ramo automóvel, no sentido de melhor compreender algumas das necessidades de otimização dos seus processos de funcionamento. Como resultado, foi identificado a necessidade de uma plataforma de gestão, que permita ao dono da oficina gerir as marcações dos seus clientes e que estes possam, através de uma aplicação móvel, solicitar pedidos de marcações e visualizar o histórico de intervenções nas suas viaturas.

Foi efetuada uma pesquisa de mercado por aplicações que pudessem suprir as necessidades das oficinas, e apesar das plataformas existentes já proporcionarem um bom desempenho, verifica-se a inexistência de um ecossistema, no qual seja possível ao dono da oficina gerir todo o processo de marcações e ao utilizador ter acesso a informações relativas às diversas intervenções nas suas viaturas, a partir do seu smartphone.

Assim, foi desenvolvida uma aplicação móvel, um website e uma plataforma de gestão, destinados a oficinas do ramo automóvel, utilizando tecnologias web e móveis da atualidade. O principal objetivo da plataforma e aplicação móvel desenvolvidas, e aqui descritas nesta dissertação, passa, não só pelo apoio na gestão de marcações de uma oficina automóvel, mas também, pelo proporcionar de uma maior ligação entre um cliente e uma oficina, através de uma aplicação móvel específica para o efeito.

Durante o desenvolvimento, foi possível verificar através de testes de usabilidade e de experiência do utilizador, realizados a potenciais utilizadores da plataforma e da aplicação móvel, o sucesso das mesmas, junto destes, potenciando uma maior aproximação entre o cliente e a oficina.

ABSTRACT

Nowadays the smartphone market has been growing in a very significant way, providing that each citizen can own at least one of these devices.

In order to develop the project described in this dissertation, several automotive workshops were contacted in order to better understand their needs, at the level of optimization of its operating processes. As a result, it has been identified the need for a management platform that allows the workshop owner to manage the schedules of its customers and to enable them through a mobile application to request schedules as well as to consult the history of repair and service in their vehicles.

A market research was carried out on the applications that could meet the needs of the workshops, and although the existing platforms already provide good performance, there is no ecosystem, with which it is possible for the workshop owner to manage the entire process schedules and the user have access to information about the various repair and service operations in their vehicles, from their smartphone.

Thus, a mobile application, a website and a management platform, were developed for automotive workshops using web and mobile technologies. The main objective of the platform and mobile application, was the support of schedule management, but also providing a greater connection between a client and the workshop, through a specific mobile application for the purpose.

During the development, it was possible to verify, through usability and user experience tests, the suitability of the developed products, enabling a closer relationship between the customer and the automotive workshop.

ÍNDICE

1. INTRODUÇÃO	1
1.1 CONTEXTUALIZAÇÃO	1
1.2 MOTIVAÇÃO	2
1.3 PROBLEMA	2
1.4 OBJETIVOS	3
1.5 ESTRUTURA DA DISSERTAÇÃO	3
2. ESTADO DA ARTE.....	5
2.1 INTRODUÇÃO	5
2.2 IDENTIFICAÇÃO DE NECESSIDADES.....	5
2.3 APLICAÇÕES EXISTENTES NO MERCADO	5
2.4 TECNOLOGIAS DE DESENVOLVIMENTO	11
2.4.1 ARQUITETURAS DE DESENVOLVIMENTO DE SOFTWARE	11
2.4.2 TECNOLOGIAS DE DESENVOLVIMENTO DESKTOP	13
2.4.3 TECNOLOGIAS DE DESENVOLVIMENTO WEB	14
2.4.4 TECNOLOGIAS DE DESENVOLVIMENTO PARA DISPOSITIVOS MÓVEIS	17
2.5 UMA POSSÍVEL SOLUÇÃO PARA O PROBLEMA.....	18
2.6 CONCLUSÃO	19
3. PROJETO DE UMA SOLUÇÃO	21
3.1 INTRODUÇÃO	21
3.2 VISTA DA ARQUITETURA	21
3.3 REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS.....	22
3.4 ARQUITETURA E ESTRUTURA DOS SISTEMAS.....	24
3.5 CASOS DE USO DAS APLICAÇÕES WEB E MOBILE	26
3.6 MOCKUPS DAS APLICAÇÕES WEB E MÓVEL	27
3.7 MODELO DE DADOS.....	32
3.7 CONCLUSÃO	35
4. DESENVOLVIMENTO DA PROPOSTA	36
4.1 A FRAMEWORK DE DESENVOLVIMENTO LARAVEL	36
4.2 A FRAMEWORK DE DESENVOLVIMENTO IONIC	37

4.3 DESENVOLVIMENTO E IMPLEMENTAÇÃO DA APLICAÇÃO WEB	37
4.3.1 ESTRUTURA	38
4.3.2 MÓDULOS DA APLICAÇÃO WEB	40
4.4 DESENVOLVIMENTO E IMPLEMENTAÇÃO DA APLICAÇÃO MÓVEL	56
4.4.1 ESTRUTURA	56
4.4.2 MÓDULOS DA APLICAÇÃO MÓVEL	58
4.6 MANUTENÇÃO DAS PLATAFORMAS DE DESENVOLVIMENTO	66
5. TESTES E ANÁLISE DE RESULTADOS	68
5.1 TESTES DE USABILIDADE	68
5.2 TESTES DE EXPERIÊNCIA DO UTILIZADOR E VALIDAÇÃO DE REQUISITOS	73
5.3 CONCLUSÃO	75
6. CONCLUSÕES E PERSPETIVAS DE DESENVOLVIMENTO FUTURO	76
REFERÊNCIAS	79
ANEXOS	82

ÍNDICE DE FIGURAS

FIGURA 1 - SOFTWARE ARTSOFT	6
FIGURA 2 - SOFTWARE OFFICEGEST	7
FIGURA 3 - SOFTWARE XD SOFTWARE	8
FIGURA 4 - ESQUEMA MVVM	11
FIGURA 5 - ESQUEMA THREE TIER.....	12
FIGURA 6 - ESQUEMA MVC	12
FIGURA 7 - VISTA DA ARQUITETURA	21
FIGURA 8 - ARQUITETURA DO SISTEMA	24
FIGURA 9 - ESTRUTURA GERAL DAS APLICAÇÕES WEB E MÓVEL.....	25
FIGURA 10 - CASO DE USO PLATAFORMA DE GESTÃO.....	26
FIGURA 11 - CASO DE USO APLICAÇÃO MÓVEL	27
FIGURA 12 - MOCKUP PLATAFORMA WEB: LOGIN	28
FIGURA 13 - MOCKUP PLATAFORMA WEB: AUTOMÓVEIS	28
FIGURA 14 - MOCKUP PLATAFORMA WEB: PAINEL	29
FIGURA 15 - MOCKUP PLATAFORMA WEB: CLIENTES	29
FIGURA 16 - MOCKUP APLICAÇÃO MOBILE: LOGIN.	30
FIGURA 17 - MOCKUP APLICAÇÃO MOBILE: MENU.....	31
FIGURA 18 - MOCKUP APLICAÇÃO MOBILE: MEUS CARROS.	31
FIGURA 19 - BASE DE DADOS	34
FIGURA 20 - ESTRUTURA DO CÓDIGO DA PLATAFORMA WEB.....	38
FIGURA 21 - PLATAFORMA WEB: LOGIN	42
FIGURA 22 - PLATAFORMA WEB: GESTÃO DE UTILIZADORES	44
FIGURA 23 - PLATAFORMA WEB: GESTÃO DE AUTOMÓVEIS.....	45
FIGURA 24 - PLATAFORMA WEB: GESTÃO DE SERVIÇOS.	46
FIGURA 25 - PLATAFORMA WEB: GESTÃO DE ARTIGOS	46
FIGURA 26 - PLATAFORMA WEB: GESTÃO DE MARCAÇÕES	47
FIGURA 27 - PLATAFORMA WEB: PAINEL	48
FIGURA 28 - PLATAFORMA WEB: REGISTO DE ATIVIDADE	49
FIGURA 29 - PLATAFORMA WEB: GESTÃO DE ADMINISTRADORES	50
FIGURA 30 - PLATAFORMA WEB: API	52
FIGURA 31 - PLATAFORMA WEB: HOME.....	53
FIGURA 32 - PLATAFORMA WEB: GESTÃO DE AUTOMÓVEIS PARA VENDA	54

FIGURA 33 - PLATAFORMA WEB: MENSAGENS.....	55
FIGURA 34 - ESTRUTURA DO CÓDIGO DA APLICAÇÃO MOBILE.....	56
FIGURA 35 - APLICAÇÃO MOBILE: LOGIN	60
FIGURA 36 - APLICAÇÃO MOBILE: PERFIL	61
FIGURA 37 - APLICAÇÃO MOBILE: MEUS CARROS	62
FIGURA 38 - APLICAÇÃO MOBILE: SERVIÇOS	63
FIGURA 39 - APLICAÇÃO MOBILE: NOVA MARCAÇÃO	64
FIGURA 40 - APLICAÇÃO MOBILE: NOTIFICAÇÃO E LEMBRETES	65
FIGURA 41 - ESQUEMA NOTIFICAÇÕES.....	66
FIGURA 42 - GRÁFICO DE RESULTADOS SUS	70
FIGURA 43 - RESULTADO SUS PERGUNTA 1	71
FIGURA 44 - RESULTADO SUS PERGUNTA 5	71
FIGURA 45 - RESULTADO SUS PERGUNTA 6	72
FIGURA 46 - RESULTADO SUS PERGUNTA 8	72
FIGURA 47 - PLATAFORMA WEB: HOME.....	82
FIGURA 48 - PLATAFORMA WEB: HOME CONTINUAÇÃO	83
FIGURA 49 - PLATAFORMA WEB: SOBRE NÓS.....	83
FIGURA 50 - PLATAFORMA WEB: DETALHES AUTOMÓVEL.....	84
FIGURA 51 - PLATAFORMA WEB: DETALHES AUTOMÓVEL CONTINUAÇÃO	84
FIGURA 52 - PLATAFORMA WEB: SERVIÇOS	85
FIGURA 53 - PLATAFORMA WEB: CONTACTOS	85
FIGURA 54 - PLATAFORMA WEB: ADICIONAR NOVO CLIENTE	86
FIGURA 55 - PLATAFORMA WEB: DETALHES DE CLIENTE.....	86
FIGURA 56 - PLATAFORMA WEB: APAGAR CLIENTE	87
FIGURA 57 - PLATAFORMA WEB: REGISTO VEÍCULO	87
FIGURA 58- PLATAFORMA WEB: MARCAÇÕES (CALENDÁRIO)	88
FIGURA 59 - PLATAFORMA WEB: ADICIONAR NOVO AUTOMÓVEL PARA VENDA	88

ÍNDICE DE TABELAS

TABELA 1 - COMPARAÇÃO ENTRE APLICAÇÕES.	9
TABELA 2 - ALGUMAS VANTAGENS E DESVANTAGENS DAS APLICAÇÕES DESKTOP.	14
TABELA 3 - ALGUMAS VANTAGENS E DESVANTAGENS DA APLICAÇÃO WEB.	15
TABELA 4 - COMPARAÇÃO ENTRE DIFERENTES FRAMEWORKS.	16
TABELA 5 - APLICAÇÕES HÍBRIDAS VS. APLICAÇÕES NATIVA.	18
TABELA 6 - REQUISITOS FUNCIONAIS DA APLICAÇÃO WEB E MOBILE.	22
TABELA 7 - TESTES APLICAÇÃO MÓVEL	73
TABELA 8 - SUGESTÕES E COMENTÁRIOS DOS PARTICIPANTES DO QUESTIONÁRIO	74

LISTA DE SIGLAS E ABREVIATURAS

IUC - Imposto Único de Circulação

API - Application Programming Interface

JWT-AUTH - JSON Web Token Authentication

MVC - Model View Controller

ORM - Object Relational Mapping

URL - Uniform Resource Locator

HTTP - Hypertext Transfer Protocol

XML - eXtensible Markup Language

JSON - JavaScript Object Notation

CRUD - Create, read, update and delete

JS - JavaScript

MVVM - Model–View–ViewModel

1. INTRODUÇÃO

Com o desenvolvimento desta aplicação móvel, website e plataforma de gestão de oficinas pretende-se dar a conhecer todo o planeamento antecedente e o trabalho desenvolvido, decisões das melhores tecnologias a serem usadas e todo o processo para a concretização destas. Em primeiro lugar, falar-se-á da motivação que levou a desenvolver este projeto para a dissertação de mestrado, seguindo-se a contextualização da dissertação e o problema que se pretende resolver com o culminar deste software. Sucessivamente será feita uma apresentação dos objetivos que envolvem o pressuposto desta dissertação. Por fim, é explicada a estrutura do documento, o mais fiel possível, para proporcionar uma melhor leitura.

1.1 Contextualização

Este trabalho ambicionou desenvolver um sistema como um todo, que seja integrado na gestão de uma oficina de automóveis com a inclusão de um website, plataforma de gestão de oficinas e uma aplicação móvel para o cliente acompanhar as várias informações/etapas do seu carro. Na conceção e desenvolvimento desta dissertação, prevaleceu a ideia, após uma primeira pesquisa, da pouca oferta no mercado deste tipo de aplicação e as existentes estarem desatualizadas e com tecnologias antigas. Ao desenvolver a dissertação espera-se poder proporcionar uma nova experiência e uma maior ligação entre o cliente e a oficina. Esta plataforma de gestão e aplicação móvel insere-se no setor profissional de oficinas do ramo automóvel. A plataforma é intuitiva e de simples uso, proporcionando uma nova experiência à oficina e ao consumidor final. Uma das maiores vantagens tem a ver com o facto de o cliente poder consultar, tanto via web como mobile, o estado do seu veículo, serviços que a oficina disponibiliza, notificações de revisões, inspeção, IUC, assim como, o seguro. Este sistema tem várias funcionalidades entre as quais gestão de clientes, gestão de carros e marcações.

1.2 Motivação

Várias razões levaram à concepção e desenvolvimento desta dissertação. Todo o ser humano continua a sua busca incansável e, no mundo da tecnologia cada vez mais, se sente esta necessidade de procurar novos desafios e novas ferramentas. Considerando o aumento do uso de internet e o crescimento nas vendas de smartphone (Statística, 2019), cada vez mais as pequenas e médias empresas estão a seguir esta tendência. Pretende-se assim, uma aplicação móvel eficaz, que possa envolver mais e interagir com o cliente e a oficina, do que apenas um mero site responsivo.

1.3 Problema

Este documento e dissertação de mestrado, parte do pressuposto que com a pesquisa realizada não foi encontrado no mercado uma plataforma de gestão de oficinas com todas as funcionalidades e interações que é proposto desenvolver e o qual é apresentado ao longo do documento.

De forma a comprovar esta lacuna foi feito um estudo presente no capítulo dois deste documento, de várias aplicações no ramo automóvel e chegou-se às seguintes conclusões:

- Maior parte dos sistemas de gestão deste ramo são antigos e desenvolvidos em linguagens menos usadas.
- Não foi encontrado um software que proporcionasse à oficina um website, plataforma de gestão e aplicação móvel.
- Algumas não permitem que o utilizador/cliente da oficina possa ter acesso à plataforma para poder visualizar, por exemplo, os serviços do seu carro.
- Nas aplicações analisadas/comparadas, algumas têm um sistema de alerta ao cliente, o qual é feito através de uma SMS, somando custos externos.
- Alguns dos softwares desenvolvidos neste tipo de linguagens não permitem o seu uso fora da rede interna da empresa.

1.4 Objetivos

A dissertação tem como objetivo principal desenvolver uma plataforma de gestão de oficinas, um website e uma aplicação móvel ligadas entre si com o uso de uma *API*.

Pretende-se com a conclusão desta dissertação, que seja possível facilitar a ligação entre a oficina e os seus clientes (Kappel, 2018); possibilitar ao cliente, de forma fácil e através da aplicação, agendar uma marcação a um automóvel específico; seja possível a um administrador fazer a gestão de toda a plataforma, quer seja clientes, automóveis, entre outros no *back-office* da mesma, e que ambas as aplicações desenvolvidas sejam apelativas e de fácil uso.

Por fim pretende-se testar a usabilidade e interação do utilizador com os softwares e fazer a análise dos resultados obtidos.

1.5 Estrutura da Dissertação

O documento encontra-se estruturado e organizado de forma a permitir uma compreensão progressiva.

O documento encontra-se dividido pelos seguintes capítulos:

O primeiro capítulo: Introdução. Tem como intenção dar a conhecer a dissertação, motivação, contextualização, problema e apresentação dos objetivos do trabalho.

O segundo capítulo: Estado da arte. Tem como fundamento apresentar o estudo feito na área; é realizada uma investigação no sentido de apresentar plataformas existentes no ramo de sistemas de gestão de carros; é também apresentada uma comparação das funcionalidades destas três aplicações já existentes e a plataforma que se pretende desenvolver.

O terceiro capítulo: Projeto de uma solução. Tem como propósito demonstrar e analisar todo o trabalho e estudo prévio ao desenvolvimento das duas soluções apresentadas ao longo deste documento.

O quarto capítulo: Desenvolvimento do projeto. Tem como objetivo dar a conhecer todos os passos realizados, estrutura, módulos desenvolvidos que levaram à realização da plataforma web e aplicação móvel.

O quinto capítulo: Testes e análise e resultados. Tem como intuito a análise e resultados obtidos dos testes feitos a ambas as aplicações.

O sexto capítulo: Conclusões. Tem como intenção refletir sobre o trabalho realizado e perspectivas de desenvolvimento futuro.

2. ESTADO DA ARTE

2.1 Introdução

Com este estudo do estado da arte pode-se identificar as fragilidades de projetos desenvolvidos nesta área, dando assim uma melhor percepção daquilo que poderá ser melhorado ou mesmo, desenvolver um projeto que não exista no mercado. Este estudo permitirá, também, fazer o levantamento de requisitos e estabelecer o primeiro contacto com o tema em questão. Feita esta análise, é possível assim, saber em concreto as abordagens a fazer ao desenvolver o trabalho, poupando tempo no futuro com investigações desnecessárias.

Assim sendo e, após a identificação das necessidades, foi realizada comparação entre as funcionalidades de outras três aplicações já existentes na área, bem como a apresentação destas.

2.2 Identificação de necessidades

Partindo de uma análise e visita *in loco* a quatro oficinas do ramo automóvel, constatou-se que não são usadas aplicações móveis, por parte das oficinas e respetivos clientes.

Após uma pesquisa intensiva, com entrevistas aos proprietários e consultas aos sistemas das mesmas, chegou-se à conclusão que, a maior parte das oficinas utilizam programas desatualizados. Para além disso, os mesmos não possuem todas as funcionalidades e potencialidades que estas necessitam.

Nestas mesmas visitas, para além dos requisitos recolhidos para a aplicação, conclui-se que não têm um conhecimento aprofundado da importância destas ferramentas.

2.3 Aplicações existentes no mercado

Decidiu-se fazer uma pesquisa de alguns softwares semelhantes de gestão de oficinas. Começou-se por analisar vários tipos de aplicações e devidas

funcionalidades, no entanto foi opção, pelo facto de estas serem mais as orientadas e com funcionalidades semelhantes ao pretendido, fazer-se a abordagem apenas às três que iremos destacar, ressaltando as mais indicadas para este documento.

Artsoft Business Software

A ARTSOFT (ARTSOFT, 2018) é uma empresa especializada no desenvolvimento e comercialização de soluções tecnológicas de apoio à gestão empresarial. Após a análise a este software conclui-se que este tem algumas funcionalidades chave entre as quais: sistema de autenticação, gestão de utilizadores, gestão de automóveis, gestão de serviços e alertas por email.

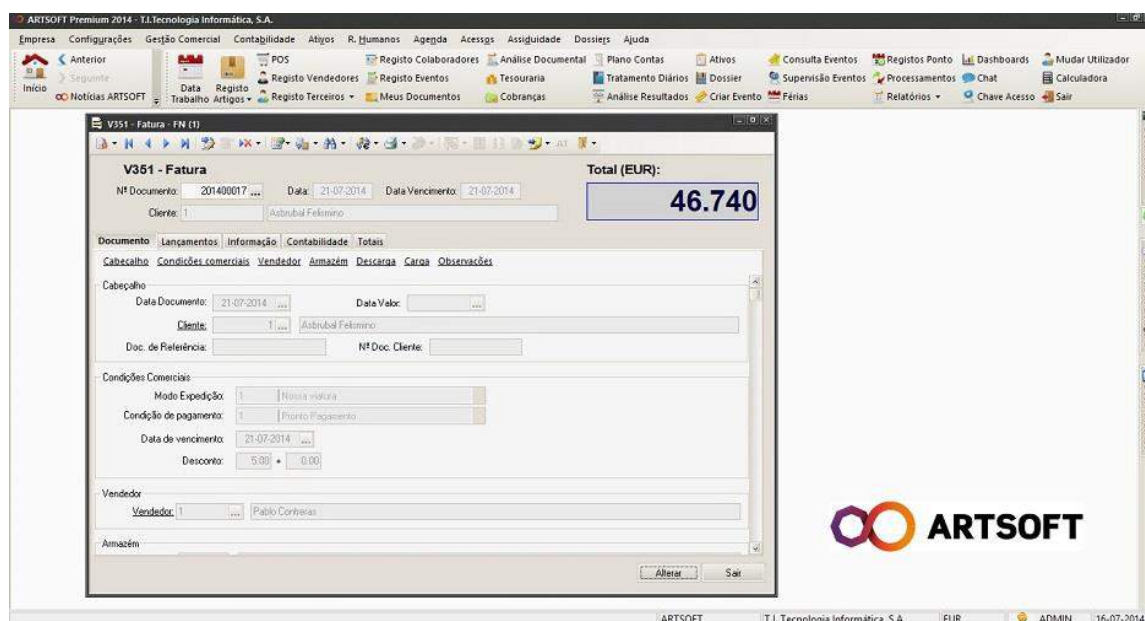


Figura 1 - Software Artsoft

O facto deste software não ser uma aplicação baseada na web, verifica-se uma falha no seu uso. Com a evolução das tecnologias, é positivo que uma aplicação de gestão deste setor seja “móvel”: usada em qualquer lugar pelo responsável do sistema.

OfficeGest

O Officegest é um produto da empresa Officegest LDA (Officegest, 2018), de capitais 100% nacionais, que há vários anos desenvolve produtos para as áreas das novas tecnologias. Após a análise desta aplicação conclui-se os seguintes pontos chave: aplicação baseada na web, mobile *friendly*, sistema de autenticação, gestão de utilizadores, gestão de automóveis, gestão de serviços, alertas por email.

The screenshot displays the OfficeGest web application interface for creating a new invoice ('Novo documento: Factura'). The interface is divided into a sidebar on the left with navigation links (Vendas, Compras, etc.) and a main content area. The main area contains several sections: 'Veículo' with fields for license plate, version, brand, year, and model; 'Entidades' for owner information; 'Linhas Doc.' table for line items; and a 'TOTALS' summary section on the right. The 'TOTALS' section shows a total liquid amount of 10.00 €, a total liquid amount of 10.00 €, a total liquid amount of 10.00 €, a total liquid amount of 10.00 €, a total liquid amount of 10.00 €, and a total liquid amount of 10.00 €. At the bottom, there are three buttons: 'Gravar Rascunho', 'Guardar e Finalizar', and 'Guardar e Imprimir'.

Figura 2 - Software OfficeGest

Este software é bastante completo, e sobretudo baseado na web. No entanto, falha, pois não permite que um cliente possa aceder à plataforma para ver dados. Outro falha, é a aplicação móvel, em que esta permita uma maior ligação entre o cliente e a oficina e, cujos alertas possam ser em tempo real.

XD Software

A génese da XD Software (XD Software, 2018) parte do objetivo de criar a plataforma tecnológica de ponto de venda mais evoluída do mercado, aliada a uma interface intuitiva e personalizável. Algumas funcionalidades chave deste

software: sistema de autenticação, gestão de utilizadores, gestão de veículos, gestão de serviços e marcações.

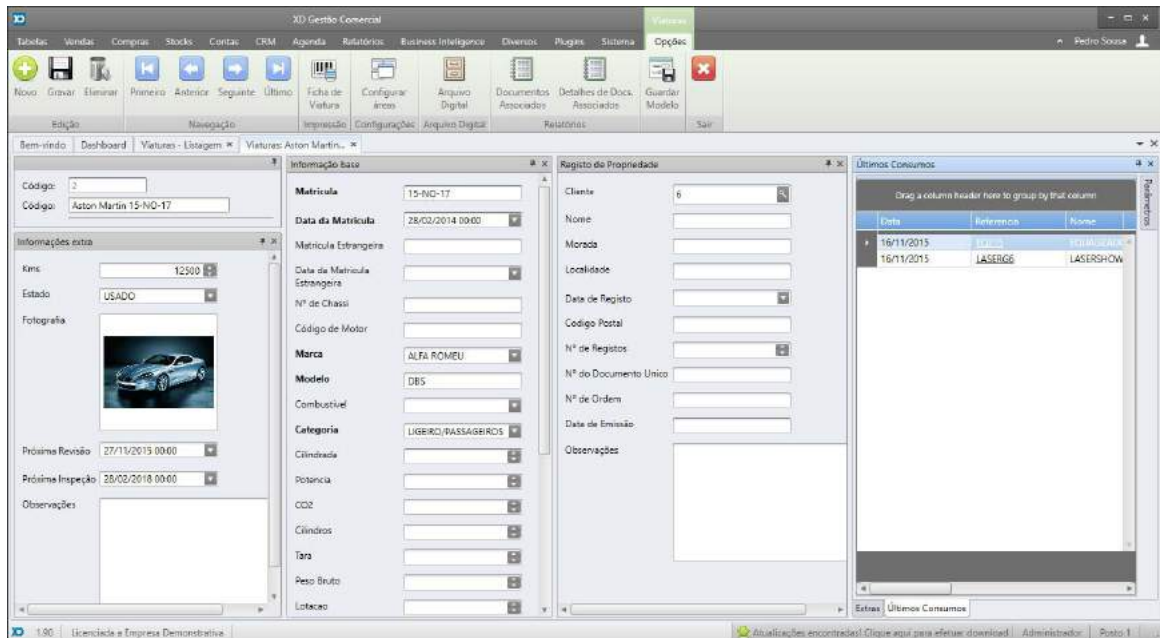





Figura 3 - Software XD Software

Este software tem também bastantes funcionalidades, mais uma vez, tal como o Artsoft, não é vantajoso por ser uma aplicação desktop.

De forma a completar a análise a estes três softwares e a verificar se estas respondiam ao problema identificado de gestão de oficinas, bem como se cumpriam os requisitos recolhidos nas oficinas visitadas, criou-se esta tabela com parâmetros relevantes de serem estudados de forma a verificar até que ponto estas aplicações suportam esta análise:

Tabela 1 - Comparação entre aplicações.

	 ARTSOFT BUSINESS SOFTWARE	 officegest. EMPOWERING SOFTWARE	 XD SOFTWARE
Aplicação baseada na web	–	Sim	Sim
Aplicação mobile <i>friendly</i>	–	Sim	–
Aplicação mobile nativa	–	–	–
Website	–	–	–
Sistema de Autenticação	Sim	Sim	Sim
Gestão de utilizadores	Sim	Sim	Sim
Gestão de carros	Sim	Sim	Sim
Gestão de serviços	Sim	Sim	Sim
Marcações	–	–	Sim
Alertas	Sim	Sim	Sim
Estatísticas	–	Sim	–

(ARTSOFT, 2018; Officegest, 2018; XD Software, 2018)

Podemos agora verificar que todos estes softwares possuem a gestão básica necessária para um programa de gestão de oficinas, no entanto ficam aquém do objetivo desta dissertação a possibilidade de para além da gestão da oficina, existir um website, uma aplicação móvel e a ligação de estas duas através de uma *API*.

2.4 Tecnologias de Desenvolvimento

Atualmente existem várias tecnologias de desenvolvimento o que por sua vez facilita e poderão agradar a qualquer pessoa que pretenda desenvolver as mais variadas aplicações. No entanto, também faz com que a escolha seja mais complicada, havendo uma vasta gama de oferta. Neste subcapítulo pretende-se abordar o desenvolvimento de aplicações recorrendo a tecnologias desktop, web e móveis, dar a conhecer algumas vantagens e desvantagens das mesmas e, também, abordar e explicar as tecnologias predominantes.

2.4.1 Arquiteturas de desenvolvimento de software

O desenvolvimento de aplicações web requer todo o cuidado possível, mas também a simplificação deste. Para tal existem vários tipos de arquitetura de desenvolvimento, como por exemplo, o *MVVM*, o *Three Tiers* ou o *Model View Controller*.

O MVVM é um padrão de arquitetura que realça a separação de interesses, permite separar a lógica da interface do utilizador da lógica do *back-end*. Tem como princípio: “*Manter o código da interface do utilizador simples e livre da lógica da aplicação para facilitar a sua gestão*” (Saleh, 2017). Este padrão separa a arquitetura em três camadas diferentes: *Model* que representa a lógica do *back-end*/lógica de negócio de dados, o *ViewModel* que interage com o *Model* e a *View* que recebe os dados de forma a possibilitar a atualização dos elementos da UI.

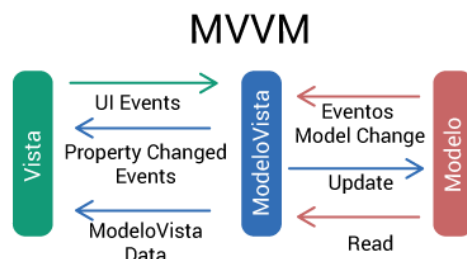


Figura 4 - Esquema MVVM

(Saleh, 2017)

A arquitetura *Three Tier* é normalmente utilizada em aplicações *client-server*. Esta é dividida em três camadas, a *Presentation Tier* é a camada de *front-end* que consiste na *user interface*, a segunda camada designa-se por *Application Tier* em que esta contém a lógica da aplicação e os seus principais recursos e, por fim, a camada *Data Tier* responsável pelo armazenamento de dados/base de dados (Tarhini, 2011).

Three Tiers

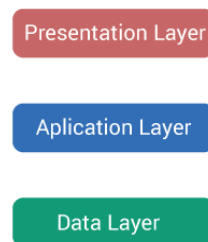


Figura 5 - Esquema Three Tier

O modelo *MVC* é um padrão de arquitetura de software (Majeed & Rauf, 2018), que separa a aplicação em três camadas distintas. A camada de interação com o utilizador (vista), a manipulação de dados (modelo) e a camada de controlo (controlador).

MVC

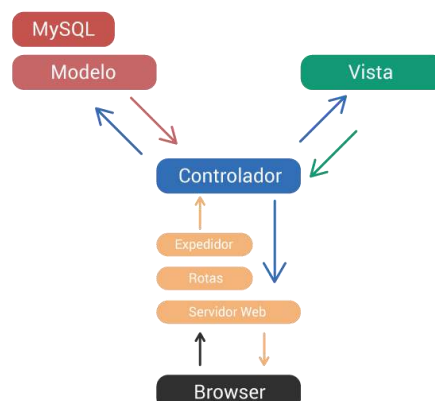


Figura 6 - Esquema MVC

O modelo representa os dados, as regras aplicadas a esses dados e ajuda a reduzir a complexidade do código que o programador precisa de escrever.

A vista é quem gere toda a parte gráfica.

O controlador processa e responde a eventos; este interage diretamente com o modelo na troca de todos os dados necessários. O controlador também interage com as *views* podendo estas assim receber informação a ser mostrada (Pop & Altar, 2014).

Será esta a arquitetura de desenvolvimento que será utilizada no desenvolvimento das aplicações, pois parece a mais indicada e também é a arquitetura predominante na maior parte das tecnologias Web.

2.4.2 Tecnologias de desenvolvimento Desktop

Existem várias tecnologias e inúmeras formas de desenvolver aplicações desktop, contudo e sendo um dos propósitos desta dissertação, que a aplicação possa ser multiplataforma uma das possibilidades seria desenvolver recorrendo a *frameworks*, visto que estas acabam por ser mais económicas e desenvolvidas em menos tempo. Como por exemplo a *Haxe*, *framework* multiplataforma de alto nível e de código aberto, possibilita compilar para uma vasta variedade de plataformas de programação, como o *Javascript*, *Flash*, *PHP*, *C++* e *NekoVm* e faz com que seja possível criar as mais variadas aplicações e até jogos. A *NW.js* permite também o desenvolvimento de software multiplataforma utilizando tecnologias web como o *HTML*, *CSS3* e *Javascript*. Outra *framework* existente é a *Electron*, criada inicialmente para o *Atom* (editor de código), que permite criar aplicações desktop multiplataforma utilizando também tecnologias Web.

Algumas vantagens e desvantagens das aplicações desktop

Pode-se agora analisar algumas vantagens e desvantagens do desenvolvimento de aplicações para desktop:

Tabela 2 - Algumas Vantagens e desvantagens das aplicações desktop.

Vantagens	Desvantagens
Não necessita de conexão à internet para a maior parte das funcionalidades.	Dificuldade na ligação da aplicação desktop com a mobile.
Dados armazenados localmente o que por vezes ajuda a evitar a perda de dados.	Necessidade de ser instalada num computador.
Maior probabilidade na estabilidade do software.	Restrita apenas aos computadores que tem instaladas a aplicação.

Depois de estudadas estas vantagens e desvantagens pode-se concluir que o desenvolvimento de uma aplicação desktop, não seria o mais viável para esta dissertação, restringindo esta ao computador em que a mesma fosse instalada, dificultando assim, o uso por vários utilizadores e, de certa forma, também seria complicado, se o software não estivesse constantemente com acesso à internet para fazer a conexão constante entre esta e a aplicação mobile.

2.4.3 Tecnologias de desenvolvimento Web

O desenvolvimento web (Doyle & Lopes, 2005) é um termo normalmente utilizado para descrever as atividades que envolvem o desenvolvimento de um website. Este desenvolvimento pode passar pela criação de uma simples página estática, como o desenvolvimento de uma rede social, fóruns, sistemas de gestão, entre outros.

Este tipo de aplicações, normalmente, é dividido em duas categorias: o desenvolvimento *back-end* e o *front-end*. Tomar decisões entre quais tecnologias utilizar não é tarefa simples, pois cada vez mais são apresentadas opções diferentes como nunca antes. Sendo assim, falar-se-á de forma geral, sobre algumas destas tecnologias, aprofundando as que foram escolhidas para o desenvolvimento desta dissertação.

Front-End

As aplicações *font-end* são aquelas que o utilizador interage diretamente com estas e tem o primeiro contacto, por isso este desenvolvimento passa

também pelo cuidado com a interação e a experiência do utilizador. O desenvolvimento por norma é recorrido às tecnologias HTML (linguagem de marcação), CSS (linguagem dos estilos) e Javascript (linguagem de programação do lado do cliente).

Back-End

O desenvolvimento *back-end*, poderá dizer-se, que é toda a lógica da aplicação; uma camada que está escondida da vista do utilizador (Red & Yellow, 2019). Este é responsável por criar todas as funcionalidades que o sistema apresenta. Todo ele passa pela criação do servidor, base de dados, *API*, entre outros, necessários para que o sistema seja executado na perfeição. Existe uma vasta gama de tecnologias que pode ser usada para o seu desenvolvimento entre as quais: *PHP*, *Java*, *Ruby* ou *Python*.

Algumas vantagens e desvantagens de aplicações web

Serão agora apresentadas algumas vantagens e desvantagens das aplicações web. A tabela 3 (Magic Web Solutions, 2019) que se segue apoiou a tomada de decisão pelo sistema Web:

Tabela 3 - Algumas vantagens e desvantagens da aplicação Web

Vantagens	Desvantagens
Aplicações são acedidas via web, não necessitam de instalação no computador.	Falta de padronização por parte dos diferentes navegadores.
Mais fácil de instalar, fazer manutenção e manter segura.	Na falta de internet a aplicação fica inacessível.
Facilidade em aumentar o poder de processamento.	Caso exista falha por parte do serviço de alojamento a aplicação fica inutilizável.
Facilidade em exportar dados da aplicação.	
Acesso via web ou mobile.	

Podemos assim concluir que o desenvolvimento de uma aplicação web será o mais viável para esta dissertação, pois facilitará o acesso por qualquer pessoa em qualquer parte desde que a mesma tenha acesso à internet. A manutenção apenas é focada e centralizada na aplicação, o que permite poupar tempo e o acesso à mesma, quer através de um computador, ou um dispositivo móvel.

Algumas comparações entre frameworks de desenvolvimento web

Na tabela abaixo apresentada pode-se verificar a comparação entre várias *frameworks* web, a sua linguagem, e se se utiliza o modelo *MVC*.

Tabela 4 - Comparação entre diferentes frameworks

Framework	Modelo MVC	Linguagem
Laravel	Sim	PHP
Express	Não	JavaScript
Phoenix	Sim	Elixir
Yii2	Sim	PHP
Ruby on Rails	Sim	Ruby
Spring	Sim	Java
Django	Não	Python

(Django, 2018; Express, 2018; Laravel, 2018; Phoenix, 2018; Ruby on Rails, 2018; Spring, 2018; Yii2, 2018)

Com se pode verificar e como já referido neste documento, o modelo *MVC* encontra-se em diferentes *frameworks* de desenvolvimento web e nas mais variadas linguagens de programação.

2.4.4 Tecnologias de desenvolvimento para Dispositivos Móveis

O desenvolvimento móvel surgiu com a criação dos telemóveis por isso esta é uma área relativamente nova quando comparando com o desenvolvimento desktop e web. Este pode ser criado para as plataformas android, iOS ou Windows *phone*. Atualmente, existem várias ferramentas para o desenvolvimento destas sendo algumas delas o Android Nativo, *Objective-C*, *Xamarin*, *Cordova*, entre outras. Este desenvolvimento pode ser feito através de aplicações nativas ou híbridas.

Aplicações Nativas

As aplicações nativas são desenvolvidas unicamente para um sistema móvel específico (Android, iOS). Estas utilizam linguagens de programação específicas, o *Objective-C* para o iOS e o Java em Android. Tem como principais características oferecerem uma experiência de utilização otimizada ao dispositivo.

Aplicações Híbridas

O desenvolvimento de aplicações híbridas utiliza maioritariamente tecnologias web (Heitotter, Hanschke, & Majchrzak, 2013), como o *HTML5*, *JavaScript*, *CSS*, entre outras. Estas por norma são desenvolvidas recorrendo ao uso de *frameworks* específicas, como o *React Native*, *Ionic* ou *Xamarin*. Uma das principais vantagens é que estas permitem que o produto final seja utilizado quer num telemóvel com ambiente Android, quer em iOS ou Windows. O ciclo de desenvolvimento por norma é substancialmente mais curto e mais barato. Para o utilizador final é invisível que uma aplicação seja desenvolvida com uma destas tecnologias.

Algumas comparações entre aplicações híbridas e nativas

Apresenta-se agora uma pequena tabela, exibindo as vantagens entre desenvolver a aplicação móvel em ambiente nativo e em ambiente híbrido.

Tabela 5 - Aplicações Híbridas vs. Aplicações Nativa

Aplicações Híbridas	Aplicações Nativas
Mais barato para desenvolver.	Melhor experiência para o utilizador final.
Consegue ter um maior alvo do mercado.	Acesso direto a características e serviços via API.
Facilidade na criação de protótipos.	Produto mais preciso para um mercado único.
Acesso à maior parte das livrarias Javascript existentes.	Mais fácil encontrar erros (<i>debug</i>).
Facilidade para programadores provenientes de programação Web.	Utilização de IDE nativo.
Multiplataforma (Android, IOS, Windows Phone).	Documentação e comunidade maior.
Menor performance	Melhor performance

(ABED, 2018; Anant & Chandran, 2018)

Atualmente, é de grande interesse o desenvolvimento de aplicações para dispositivos móveis, principalmente nas áreas económicas e científicas. Para o utilizador final, as aplicações nativas, proporcionam uma melhor experiência para o utilizador, enquanto que as aplicações híbridas tem como objetivo final obter o mesmo desempenho que as aplicações híbridas conseguindo ao mesmo tempo abranger o maior número possível de plataformas móveis diferentes (Xanthopoulos & Xinogalos, 2013) .

2.5 Uma possível solução para o problema

Após esta comparação feita às três aplicações acima referidas, e no culminar das visitas feitas às quatro oficinas, sentiu-se realmente o quão é necessário o desenvolvimento desta dissertação. Tendo-se constatado que as aplicações móveis para smartphones e tablets têm vindo a assumir uma importância crescente, tanto na promoção dos negócios, como na forma como as pessoas e as organizações interagem entre si na busca e disponibilização de informação, e ressaltando-se o facto de algumas destas plataformas já

estarem estabelecidas no mercado há alguns anos, esta que é apresentada, será uma mais valia para quem a adquirir, pois possibilitará uma maior relação entre o cliente e a oficina, o que poderá fazer a diferença na escolha do cliente entre uma oficina e outra. Assim sendo desenvolver-se-á uma plataforma como um todo, onde a oficina terá a possibilidade de ter um sistema de gestão, um website e uma aplicação móvel. Como esta plataforma estará alojada online, irá permitir o seu uso em qualquer lugar. Possibilitará, também, de uma forma restrita, o acesso do cliente da oficina à plataforma, para visualização de informações relativas aos seus automóveis. A aplicação móvel, para além de fornecer informações ao cliente, permitirá também, receber notificações em tempo real do estado do seu automóvel, ou até mesmo, a data da inspeção do automóvel.

"Get closer than ever to your customers. So close that you tell them what they need well before they realize it themselves." - Steve Jobs, Apple

2.6 Conclusão

É possível concluir que atualmente já existe alguma preocupação das pequenas e médias empresas no que respeita ao marketing e ao crescimento destas. Devido a essa preocupação, surgiu esta aplicação que oferece várias funcionalidades, que permitem facilitar algumas das tarefas da gestão da oficina, envolvendo e interagindo com o cliente e a oficina.

A solução a ser desenvolvida partiu de uma necessidade específica do mercado, dada a pouca existência deste tipo de aplicação e as existentes estarem desatualizadas e com tecnologias antigas.

Este tipo de empresas tem ainda, a necessidade de gerir os requisitos de aplicações através de uma solução de baixo custo, de elevada flexibilidade e personalização de acordo com o projeto em desenvolvimento. Necessitam ainda de uma ferramenta com um bom nível de usabilidade e de fácil aprendizagem, que fosse mais um veículo facilitador de comunicação entre os diversos intervenientes. Com base neste problema foi realizado um estudo aprofundado desta temática. Este incluiu o estudo do mercado, a análise da importância da gestão das oficinas durante o desenrolar da dissertação e uma análise das

principais tarefas que devem ser suportadas por este tipo de aplicação. Por fim foi ainda realizada uma análise de ferramentas comerciais e mecânicas. Todo este estudo foi crucial para perceber a problemática associada a este tipo de aplicação.

3. PROJETO DE UMA SOLUÇÃO

3.1 Introdução

Este capítulo tem como objetivo demonstrar e analisar todo o trabalho e estudo prévio ao desenvolvimento das aplicações apresentadas ao longo deste documento. Todos os subcapítulos que se seguem foram fundamentais para a concretização das aplicações, pois nestes foram possíveis definir a arquitetura do projeto, os requisitos funcionais e não funcionais das plataformas, o estudo de como as diferentes linguagens de programação comunicam entre si como poderemos verificar no subcapítulo de fluxogramas de tecnologias e aplicações, casos de uso, modelos de dados e *mockups* das aplicações.

3.2 Vista da Arquitetura

Existem várias possibilidades de escolha para a arquitetura deste software de gestão. No entanto, a escolha recai num sistema partilhado de alojamento web. Esta escolha deve-se ao facto de se ter como ambição desenvolver uma plataforma web. A infraestrutura desta arquitetura será apoiada por máquinas Linux (*CloudLinux*) ligadas através da rede interna. Existirão servidores que farão a ligação com o alojamento da base de dados e respetivos ficheiros web da plataforma.

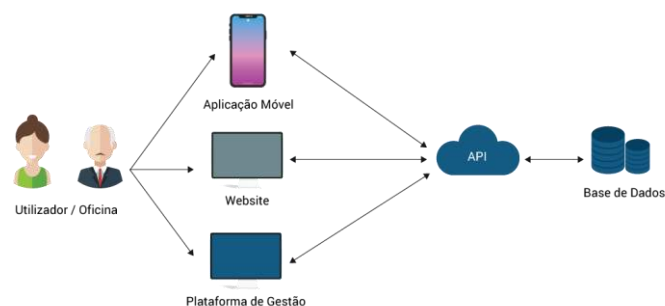


Figura 7 - Vista da Arquitetura

CloudLinux: “CloudLinux é uma distribuição Linux comercial desenhada especificamente para fornecedores de **serviços de alojamento** e **Datacenters**, permitindo tirar o máximo partido de um servidor.” («CloudLinux», 2019).

Com este sistema será possível alojar a tempo inteiro todos os ficheiros necessários para que a plataforma web e *API* funcionem na sua perfeição.

3.3 Requisitos Funcionais e Não Funcionais

Requisitos Funcionais

Os requisitos funcionais são aqueles que descrevem a funcionalidade da aplicação e os seus componentes. Os sistemas são compostos por diferentes funcionalidades, podendo estas ser despertadas por ação do utilizador ou pela aplicação em si. De seguida são descritos ambos os requisitos da aplicação web e aplicação mobile na tabela 6:

Tabela 6 - Requisitos Funcionais da Aplicação Web e Mobile

Requisitos Funcionais Aplicação Web	Requisitos Funcionais Aplicação Mobile
Sistema de Autenticação	Sistema de Autenticação
Gestão de Veículos	Visualização dos Veículos
Gestão de Utilizadores	Visualização do Perfil
Gestão de Artigos	Visualização dos Serviços
Gestão de Serviços	Visualização e possibilidade de pedido de Marcações
Gestão de Marcações	Lembretes/Notificações
Gestão do Registo de Atividade	
Gestão de Administradores	
Gestão do Website (Automóveis para venda)	

Website	
API	

Requisitos Não Funcionais

Os requisitos não funcionais são essenciais quer na aplicação web quer na mobile, para que ambas sejam eficientes, rápidas, gastem poucos recursos e que sejam fáceis de utilizar. Deste modo são apresentados os seguintes requisitos: Eficiência, performance, tempo de resposta, fiabilidade, segurança, estabilidade, usabilidade e tolerância a falhas.

3.4 Arquitetura e Estrutura dos Sistemas

De forma a facilitar o processo de desenvolvimento de ambas as aplicações web e móvel, começou-se por criar de forma visual a arquitetura do sistema das aplicações, como estas interagem e comunicam entre si. Assim sendo, pode-se agora verificar na figura 8:

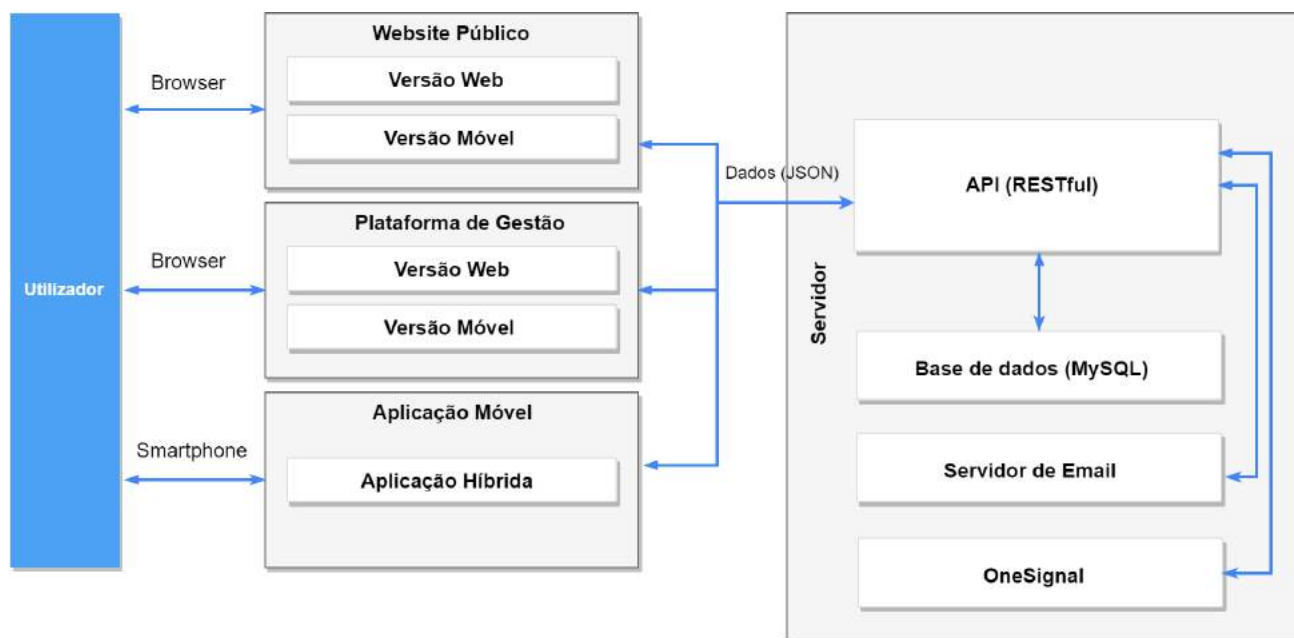


Figura 8 - Arquitetura do Sistema

Com os requisitos funcionais e não funcionais definidos sentiu-se a necessidade de desenvolver um fluxograma/esquema com todos estes requisitos de forma a facilitar e a visualizar todas as possibilidades de ambas as aplicações:

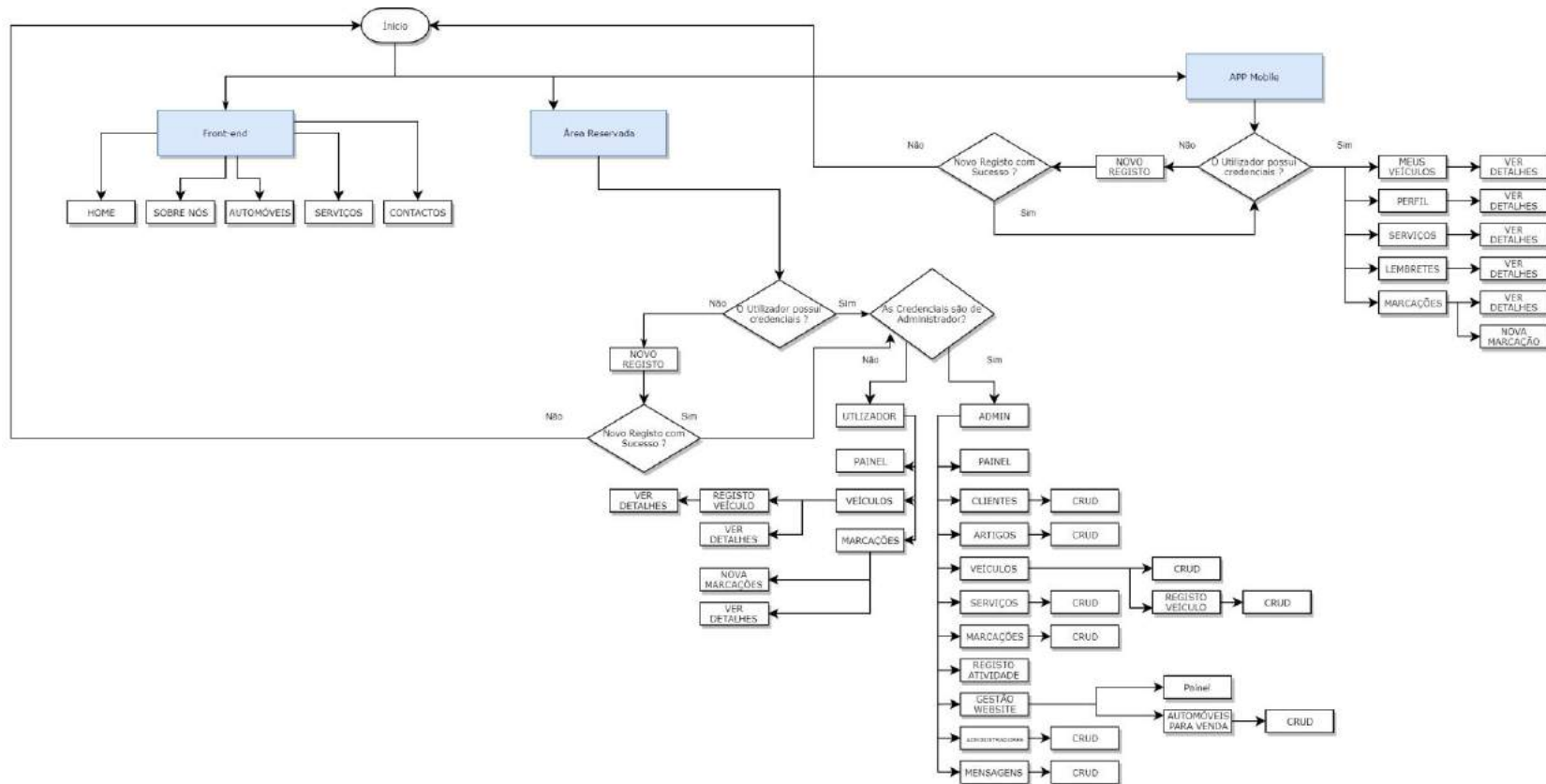


Figura 9 - Estrutura Geral das Aplicações Web e Móvel

3.5 Casos de Uso das Aplicações Web e Mobile

Os casos de uso pretendem mostrar como serão feitas as interações entre o utilizador e o sistema. Segue-se um exemplo com foco na plataforma web:

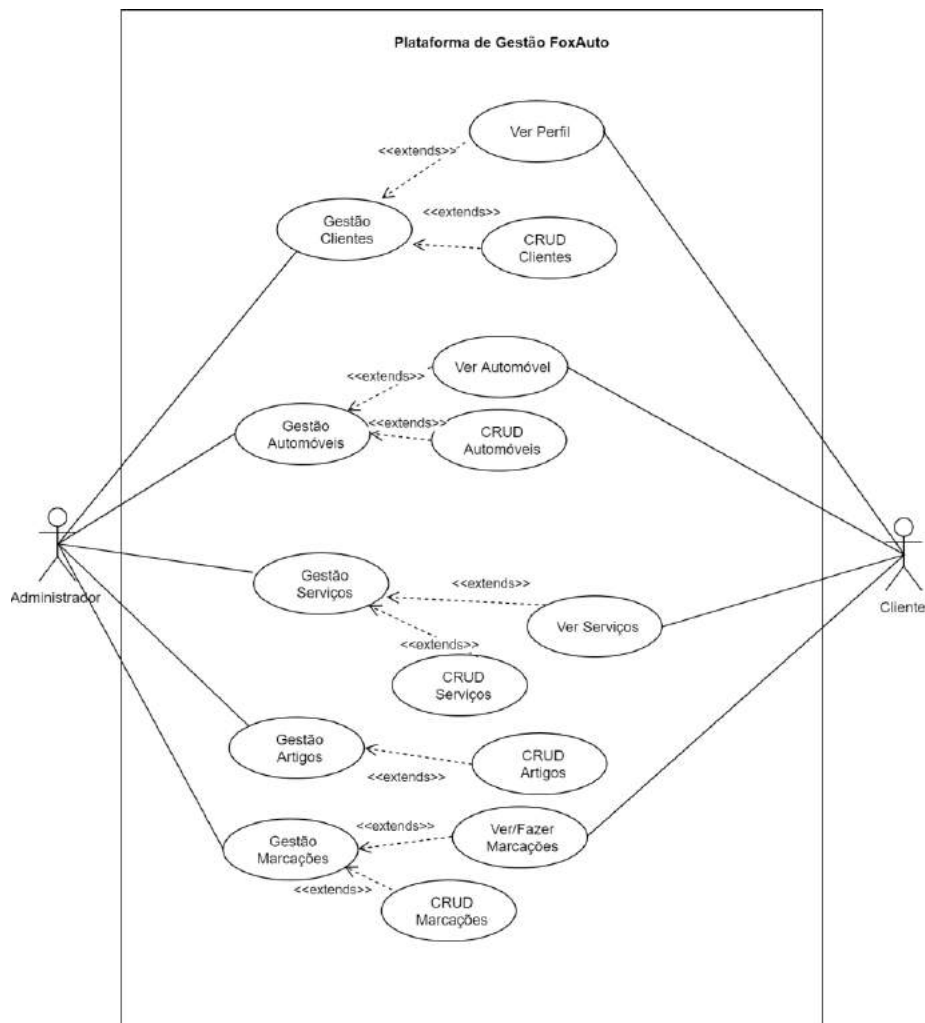


Figura 10 - Caso de Uso Plataforma de Gestão

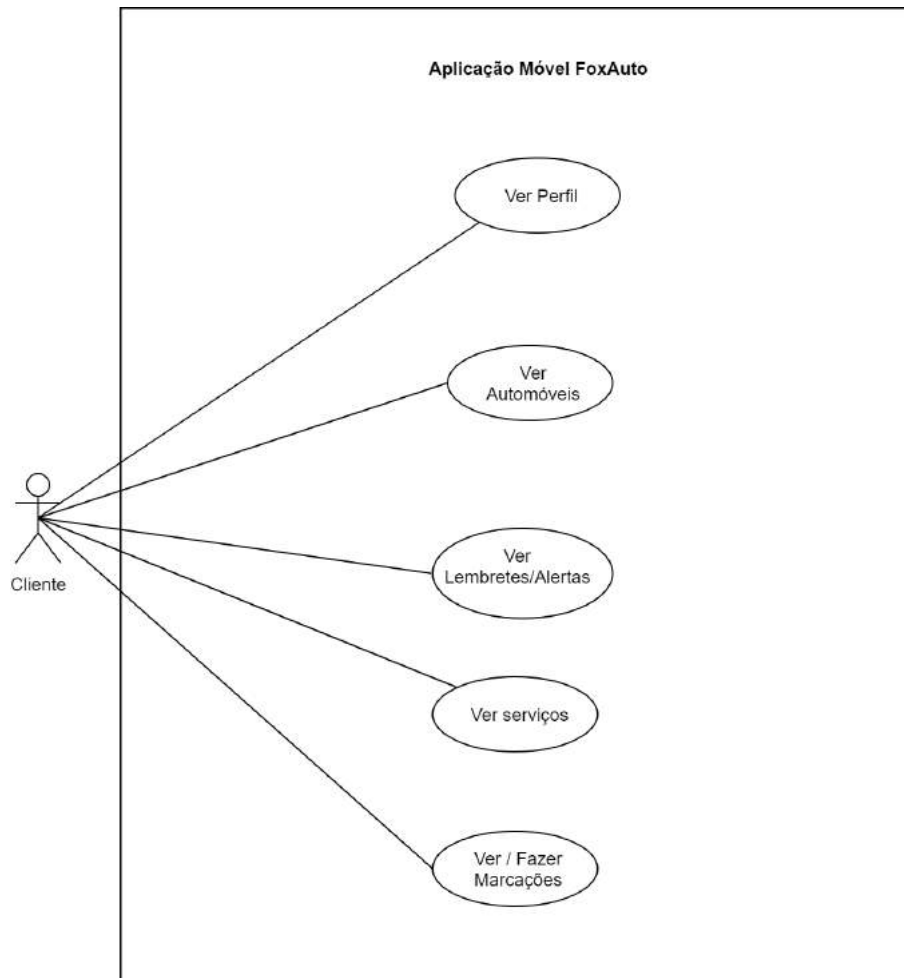


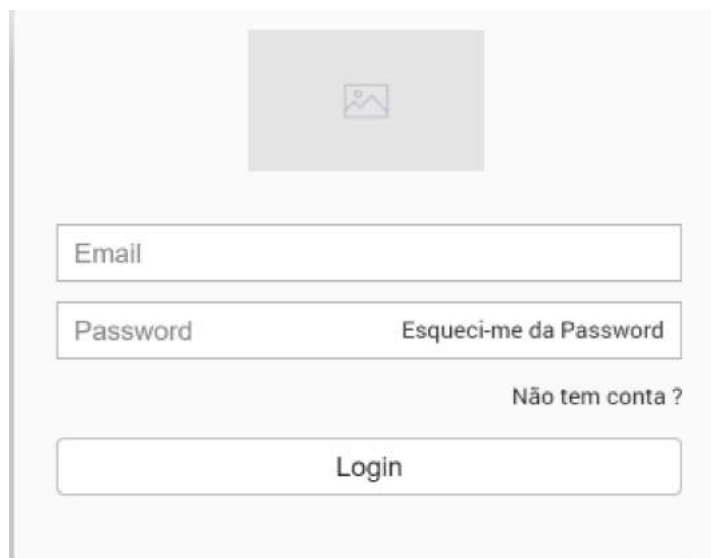
Figura 11 - Caso de Uso Aplicação Móvel

Na figura 11 pode-se verificar as ações que o cliente(utilizador) poderá realizar na aplicação móvel sendo elas: Ver os seus automóveis e registo dos mesmos; ver os lembretes e receber notificações; ver todos os serviços associados a um específico automóvel e ver e realizar uma marcação ao seu carro.

3.6 Mockups das Aplicações Web e Móvel

Para que o desenvolvimento da aplicação web fosse mais fluído e de forma a agilizar a programação das diferentes páginas, foram criados *mockups* que serviram de referência para a criação da plataforma. Apresentam-se agora alguns desses *mockups*.

Na figura 12 pode-se ver o *mockup* criado para a página de login, onde apenas o utilizador precisa de inserir o email e password para ter acesso à plataforma:



The mockup shows a login form with a header area containing a placeholder for a logo. Below the header, there is an 'Email' input field, followed by a 'Password' input field and a link labeled 'Esqueci-me da Password'. A 'Login' button is positioned below the password field. To the right of the password field, there is a link labeled 'Não tem conta ?'.

Figura 12 - Mockup Plataforma Web: Login

Mockup da página web automóveis, onde é pretendido que seja possível a pesquisa de automóveis, visualizando-se uma imagem do automóvel, modelo e descrição do mesmo:

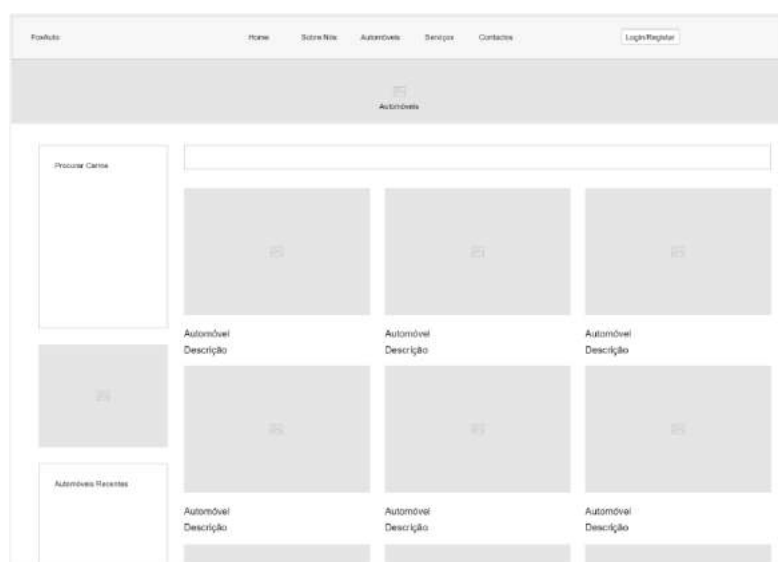


Figura 13 - Mockup Plataforma Web: Automóveis

Na figura seguinte é apresentado o *mockup* do painel de administração, sendo útil haver algumas estatísticas rápidas da aplicação bem como a aceitação de clientes novos:



Figura 14 - Mockup Plataforma Web: Painel

Nesta figura 15 pode-se ver o *mockup* da página *back-end* dos clientes, onde os dados principais destes estarão visíveis numa tabela bem como as ações que o administrador poderá aplicar a cada utilizador:

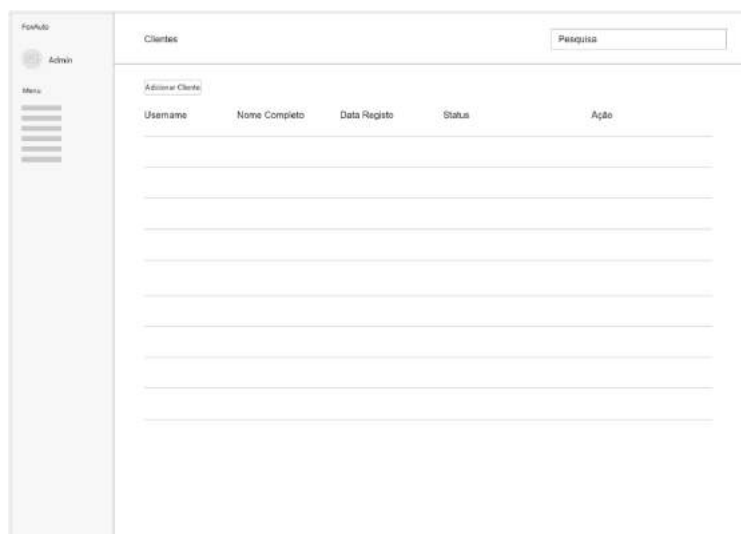


Figura 15 - Mockup Plataforma Web: Clientes

Tal como na aplicação web, antes de começar o seu desenvolvimento criaram-se *mockups* de todas as páginas da aplicação móvel de forma a ajudar à visualização do produto final e, por sua vez, poupar tempo futuro no desenvolvimento do design da mesma, sendo agora referenciados alguns desses *mockups*.

Na seguinte figura 16 pode-se verificar o *mockup* criado para a primeira interação com o utilizador, sendo esta, o login em que o utilizador tal como na aplicação web, apenas necessita de inserir o email e password para poder ter acesso à aplicação, tendo estes sido previamente registados e validados por um administrador da plataforma:

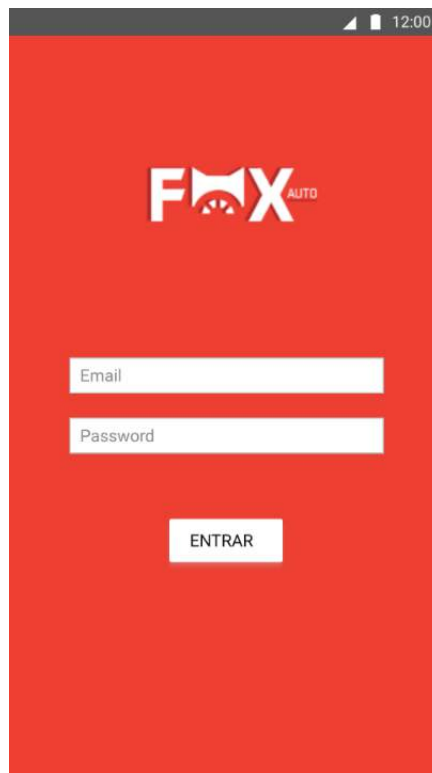


Figura 16 - Mockup Aplicação Mobile: Login.



Figura 17 - Mockup Aplicação Mobile: Menu.

Na figura 17 apresenta-se o menu da aplicação. Este apenas é acessível após o login tenha sido feito com sucesso.



Figura 18 - Mockup Aplicação Mobile: Meus Carros.

Na figura 18 pode-se ver a página 'Meus Carros', onde serão apresentados os automóveis numa lista do utilizador autenticado.

3.7 Modelo de Dados

O modelo de dados é apresentado abaixo na figura 19. Este é o responsável pelo armazenamento de toda a informação da plataforma. O esquema permite visualizar, de forma geral, as relações entre as várias tabelas, ajudando assim, a uma melhor interpretação da forma como será alojada a informação. A escolha recai sobre a tecnologia *MySQL* (MySQL, 2018), pois é uma base de dados relacional e *open source* o que faz com que não implique custos de aquisição. As vantagens da sua utilização são a velocidade de execução de tarefas, a fácil integração com o *Laravel* e por sua vez o *PHP* e o seu pouco consumo de recursos informáticos.

Segue-se uma descrição das funções das principais tabelas:

Users: Esta tabela tem como principais objetivos guardar os dados pessoais dos utilizadores da plataforma. A coluna ‘cargo’ é responsável por definir se o utilizador é administrador da plataforma ou utilizador comum. Esta tabela é imprescindível à plataforma, pois faz o elo de ligação entre as restantes tabelas da plataforma de gestão.

Carros: Tabela responsável por armazenar os dados principais dos automóveis dos utilizadores. Esta tabela faz ligação com a tabela ‘registo_carro’ e ‘marcacoes’.

Registo_Carro: Armazena mais detalhadamente os dados dos automóveis. As colunas ‘data_proxima_inspecao’, ‘data_proxima_revisao’ e ‘iuc’ para além de guardarem os dados correspondentes aos nomes delas; esta também tem outro propósito: ao serem lidas pela plataforma, será possível o sistema determinar quando é necessário enviar as notificações aos utilizadores de quando é a data da próxima inspeção/revisão, pagar o IUC ou o seguro.

Marcações: Responsável pelo armazenamento dos dados correspondentes às marcações automóveis registadas pelos utilizadores ou administrador.

Activity_Log: Regista todas as atividades feitas na plataforma, por exemplo quando o administrador altera um dado do utilizador esta tabela regista que foi feita essa alteração.

Serviços: Tabela responsável por armazenar os serviços efetuados a um determinado automóvel de um utilizador. Juntamente com a tabela 'servico_linha', 'artigos' e 'categoria' é possível saber que manutenção foi feita ao automóvel, que custo teve essa operação entre outros.

Automoveis_para_venda: Juntamente com as tabelas 'automoveis_modelo', 'automoveis_extras', 'automoveis_marca', 'automoveis_fotos' e 'extra' estas armazenam todos os dados das viaturas que o stand tem para venda e disponíveis no website.

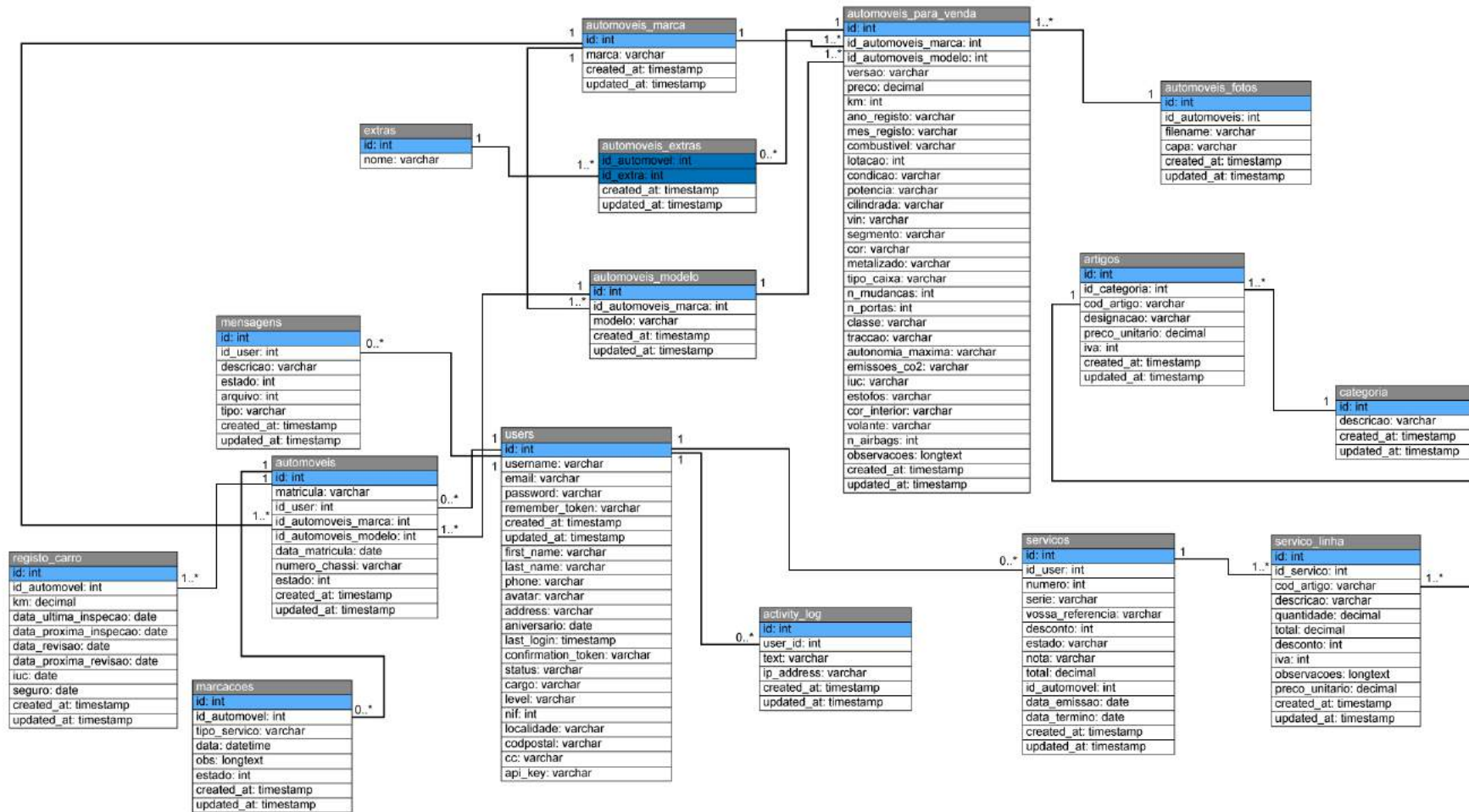


Figura 19 - Base de dados

3.7 Conclusão

Com a conclusão deste capítulo foi possível averiguar o quão é importante o planeamento inicial para o desenvolvimento de softwares. Todo este planeamento fará uma economia de tempo na criação destas aplicações, pois já foram bem definidos os requisitos funcionais e não funcionais, os *mockups* das aplicações, a arquitetura de alto nível e o modelo de dados. No capítulo seguinte será apresentado todo o desenvolvimento realizado na criação da plataforma web, aplicação móvel e *API*.

4. DESENVOLVIMENTO DA PROPOSTA

Neste capítulo pretende-se, inicialmente, apresentar as tecnologias mais importantes no desenvolvimento desta dissertação, a estrutura das aplicações e descrever detalhadamente todos os módulos que foram desenvolvidos nesta dissertação. Assim sendo, fará todo o sentido em abordar o *Laravel*, que será o *framework* principal no desenvolvimento da plataforma web de gestão e a plataforma de desenvolvimento da aplicação móvel, a *framework IONIC*.

4.1 A Framework de Desenvolvimento Laravel

O *Laravel* (Laravel, 2018) é uma *framework* de desenvolvimento rápido que utiliza como base a linguagem *PHP*. Este é livre e de código aberto. Foi criado por *Taylor B. Otwell*; utiliza o modelo de arquitetura MVC e tem como principais características ajudar a desenvolver aplicações web segura e com boa performance de forma rápida, como o código limpo e simples. Para a criação de interface gráfica, este utiliza um motor de *templates* chamado *Blade*; possui uma vasta gama de ferramentas que ajudam a criar interfaces apelativas e funcionais de forma rápida evitando a duplicação do código de programação. A comunicação entre o *Laravel* e a base de dados é feita através do *Eloquent ORM*. Este promove a boa prática ao uso de ferramentas que facilitam a inserção, atualização, pesquisa e remoção de registos da base de dados. Atualmente e aquando a elaboração deste documento a última versão deste framework é a 5.7.

O desenvolvimento da plataforma web será recorrido inteiramente à framework *Laravel*. Entre as várias frameworks de desenvolvimento web, a escolha recai sobre esta, pois parece ser a mais segura, com mais documentação e, que por sua vez, proporcionará uma melhor aprendizagem da framework.

4.2 A Framework de Desenvolvimento IONIC

Após um estudo intensivo, discernimento e tudo o que foi já abordado neste documento, foi decidido que o desenvolvimento da aplicação móvel será feito recorrendo à framework IONIC (Ionic, 2018), apesar desta utilizar tecnologias web, tem como base o *AngularJS* tecnologia à qual o autor deste documento nunca teve nenhuma interação até à data de desenvolvimento desta dissertação, sendo que é um risco, mas também um desafio que decidiu tomar. Esta framework é híbrida, ou seja, permite que ao desenvolver a aplicação esta seja compatível com o *Android*, *IOS* e *Windows Phone*. A linguagem de desenvolvimento é o *HTML5* e o *AngularJS*, o que faz com que facilite o seu uso por developers web. Como citado no site oficial do IONIC, as aplicações são criadas e desenvolvidas principalmente através da linha de comando do IONIC (o “CLI”), usa o *Cordova* para construir e executar como uma aplicação nativa (Ionic, 2018).

Segundo o estudo feito pela IBM (IBM, 2012) mostra que a melhor escolha é sempre tendo em conta o perfil da empresa e as suas necessidades e complexidades. As empresas não podem ter somente a sua estratégia, mas também, considerar o futuro do seu mercado.

Considera-se fundamental para a área em estudo a atualização ao nível dos conhecimentos científicos e informáticos. Através da partilha de experiências e saberes, da pesquisa e posterior leitura de documentos relacionados com o contexto pessoal e profissional, destaca-se o artigo na BIT magazine (Bit Magazine, 2016) sobre o impacto das aplicações para os dispositivos móveis.

4.3 Desenvolvimento e Implementação da Aplicação Web

Apresenta-se agora todas as fases do desenvolvimento da aplicação web, começando-se pela estrutura do projeto web e em seguida é apresentado e descrito todos os módulos desenvolvidos da aplicação web.

4.3.1 Estrutura

Será agora apresentada a estrutura e descrição da plataforma web. Com a apresentação desta estrutura, permitirá a qualquer pessoa entender melhor todo o trabalho desenvolvido nesta parte da dissertação. Apenas são descritos os ficheiros e diretorias essenciais e com maior relevância para este relatório. A figura 20 apresenta então a estrutura do código como se pode verificar:

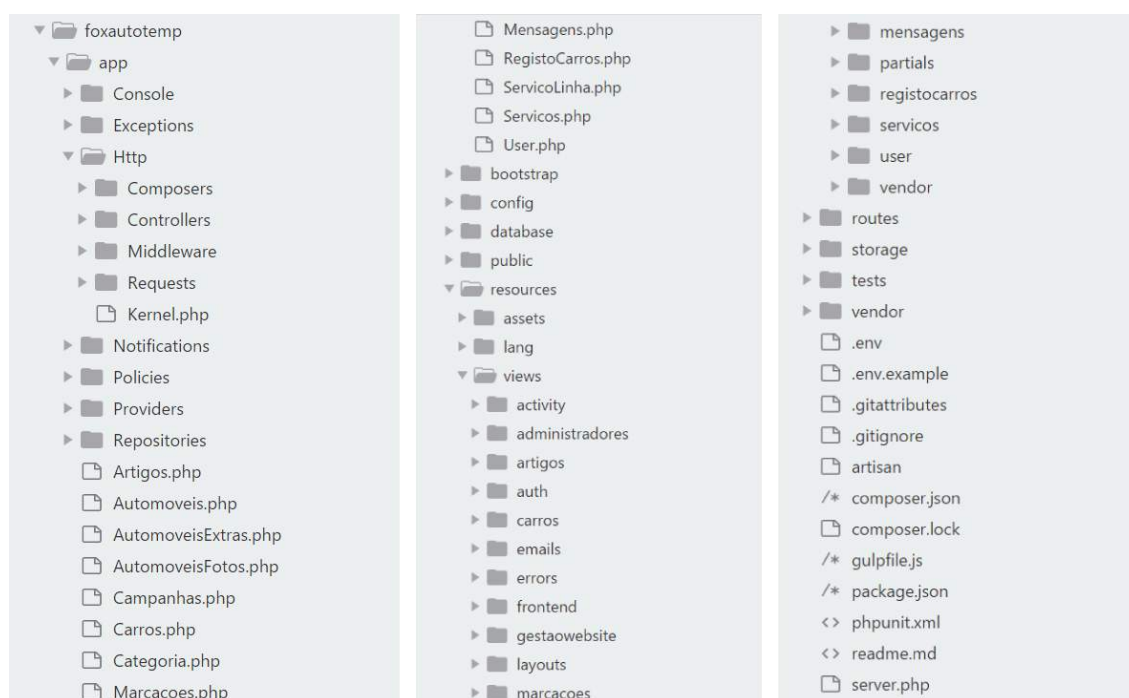


Figura 20 - Estrutura do Código da Plataforma Web

App: Diretório principal onde se encontra a maior parte do código e lógica da plataforma. Este diretório contém a maior estrutura de diretórios e subdiretórios do sistema.

Console: Pasta que contém os comandos customizados para o envio automático dos emails de aniversário, lembretes da próxima inspeção, próxima revisão, seguro e IUC.

Controllers: Esta pasta é bastante importante pois contém todos os controladores da aplicação, onde se encontra praticamente toda a lógica da plataforma.

Middleware: Pasta responsável por guardar os ficheiros responsáveis pelos filtros de requisições *HTTP* à plataforma.

Requests: Diretoria que contém todas as classes com as suas devidas validações.

Repositories: Esta pasta contém os ficheiros relativos às funções das classes da plataforma.

Config: Tal como o próprio nome indica, nesta pasta ficam os ficheiros de configuração da framework.

App.php: Ficheiro onde é definido as configurações da aplicação de forma geral, nome da aplicação, *timezone*, *debug*, *url*, *log*, *providers*, entre outros.

Mail.php: Ficheiro responsável pelas configurações dos envios de e-mail.

Jwt.php: Ficheiro com as configurações principais da api.

Public: Diretoria que contém o ficheiro principal de arranque da plataforma *index.php*. Dentro desta, temos também, subpastas com os ficheiros de fontes, imagens, *css*, *javascript*, *jquery*, entre outros.

Resources: Nesta diretoria ficam os ficheiros de visualização da aplicação, como as *views*, ficheiros de linguagem, e *assets* não compilados.

.env: Neste ficheiro são guardadas as informações sensíveis, informações de senhas e configuração do sistema. Tem como propósito manter os dados de

forma o mais segura possível. É neste ficheiro que se faz as configurações da base de dados.

.web.php: Contém todos os caminhos (*routes*) da aplicação.

4.3.2 Módulos da Aplicação Web

Nesta secção será descrita de forma o mais detalhada possível, todos os passos realizados na implementação da aplicação web, desde as linguagens de programação utilizadas aos módulos desenvolvidos. Uma vez que o pretendido seria desenvolver uma plataforma web de gestão, a escolha facilmente recaiu sobre as linguagens de programação *PHP*, *HTML* e *CSS* como base, dado conseguir dominar estas e provavelmente as mais indicadas para este projeto. Sendo assim, decidiu-se sair um pouco da zona de conforto e fazer a escolha da aprendizagem de uma nova *framework* *PHP*. Como indicado neste capítulo, a escolha recai sobre a *framework Laravel*. Sendo esta uma novidade, o início do desenvolvimento deste projeto foi bastante exaustivo e lento, pois inicialmente todo o tempo dedicado passou pela leitura exaustiva da documentação que se encontra no site da mesma. Após a sua instalação e, ao facto de se achar um pouco mais à vontade com esta nova *framework*, começou-se por fazer a ligação desta à base de dados em *MySQL* previamente desenvolvida. Este processo foi relativamente simples, pois apenas foi necessário fazer as configurações de conexão no ficheiro ‘.env’ na raiz principal do projeto. De seguida deu-se início ao desenvolvimento do módulo do sistema de login e registo.

Módulo Sistema de Login e Registo

Como primeiro passo foi desenvolvido o módulo de registo na plataforma. Este tem como função principal o utilizador poder registar-se na plataforma, definindo automaticamente como sendo utilizador da plataforma, tendo privilégios reduzidos, sendo estes diferentes do administrador. Para proceder a um novo registo foram anteriormente definidos os seguintes campos: primeiro nome, último

nome, username, email, password, confirmar password e aceitar os termos de serviço. Após este processo ser realizado, o utilizador recebe um email, enviado automaticamente pelo sistema, a confirmar o registo e a alertar que apenas poderá utilizar a plataforma após a validação na aplicação por parte de um administrador. Após a aprovação por parte de um administrador, o utilizador recebe novamente um email de aprovação e poderá então aceder à plataforma.

Será agora dado ênfase ao desenvolvimento módulo de login, que tal como no registo foi definido anteriormente que apenas eram necessários os campos de email e password para aceder à plataforma. Este módulo passou por todo o desenvolvimento de um sistema seguro e rápido. Após o login, foi programado de forma a que a plataforma automaticamente redirecione o utilizador para o painel de utilizador. No caso de ser um elemento da administração este possa ser redirecionado para o painel do mesmo. Este processo é realizado pela plataforma verificando o campo da base de dados 'cargo', caso este campo estiver preenchido como 'user' o mesmo fará login e será redirecionado para o seu devido painel. O código abaixo demonstra como é feita a autenticação de um utilizador:

```
1. public function authenticate(Request $request)
2. {
3.     $credentials = $request->only('email', 'password');
4.     $user = User::where('email', $credentials['email'])->first();
5.     $userstatus = User::where('email', $credentials['email'])->
        >where('status', '=', 'ativo')->first();
6.     if(!$user) {
7.         return response()->json([
8.             'error' => 'Email Inválido'
9.         ], 401);
10.    }
11.    if (!Hash::check($credentials['password'], $user->password)) {
12.        return response()->json([
13.            'error' => 'Password Inválida'
14.        ], 401);
15.    }
16.    if(!$userstatus) {
17.        return response()->json([
18.            'error' => 'A sua conta ainda não foi confirmada pelo Administrador.'
19.        ], 401);
20.    }
21. }
```

Nesta imagem (figura 21) é apresentada a página de Login onde o utilizador/administrador terá acesso à plataforma de gestão. Caso o utilizador não possua uma conta, poderá fazê-lo clicando no link 'Não tem uma conta?'.

Posteriormente ao registo, deverá esperar por um email da plataforma a confirmação do administrador permitindo que este utilizador aceda ao uso da plataforma.

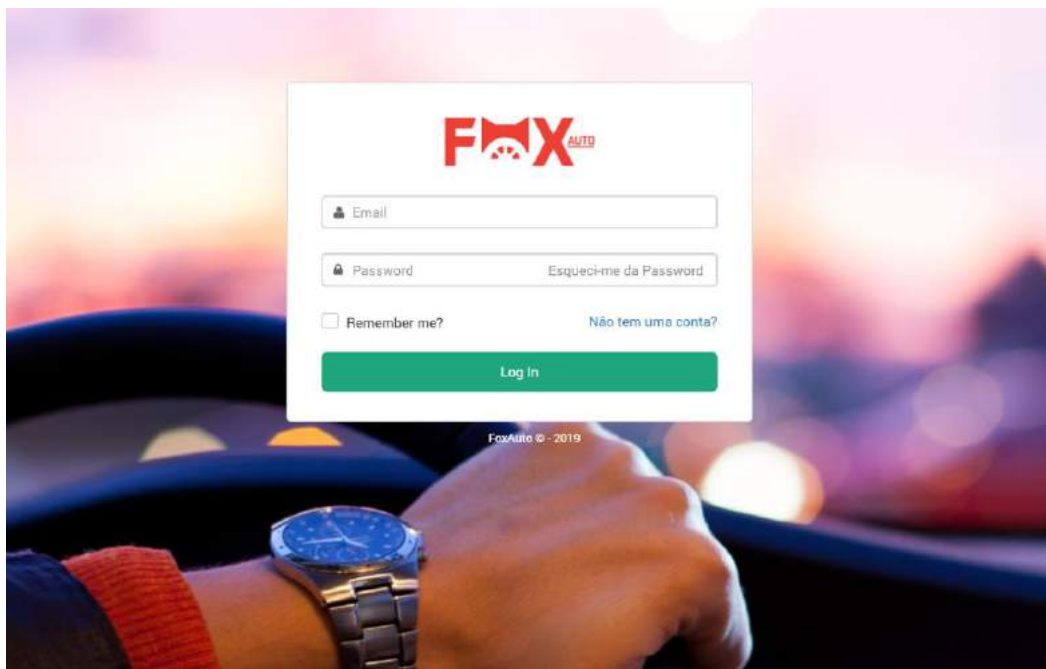


Figura 21 - Plataforma Web: Login

Em simultâneo com o desenvolvimento destes dois módulos anteriormente descritos, foi também desenvolvido o módulo de emails:

Módulo de emails

O desenvolvimento deste módulo passou pela configuração do envio automático de emails, sempre que exista um registo novo na plataforma: aniversário de um utilizador, lembretes de pagamento de IUC, seguro, inspeção, revisão e seguro do automóvel. Para ser possível o envio destes emails na plataforma foi alojado online num sistema de *Cloud Linux* como explicado anteriormente no ponto: vista de alto nível de arquitetura. Na plataforma, após o desenvolvimento da programação e lógica deste módulo, foi necessário proceder à configuração e conexão com o *system smtp* no ficheiro `‘.env’` do projeto.

Após a conclusão do desenvolvimento destes módulos anteriormente descritos, iniciou-se outro processo, ou seja, o desenvolvimento da gestão de utilizadores:

Módulo Gestão de Utilizadores

Sendo este um módulo bastante importante na plataforma, o desenvolvimento foi feito com clareza e pensado ao pormenor para que o mesmo estivesse ao mais alto nível de performance. Foi desenvolvido um sistema de CRUD, a sigla 'C' indica create em português criar, 'R' read significa a possibilidade de ler os dados, 'U' update atualizar os dados e 'D' de delete remover dados. Como tal o administrador pode adicionar novos utilizadores podendo definir o cargo deste (administrador ou utilizador), o status (confirmado ou não confirmado) e os restantes dados do utilizador definidos na tabela 'user' da base de dados. É também possível ver, editar todos os dados e apagar cada utilizador. O administrador tem a possibilidade de aceitar novos utilizadores, caso pretenda, pois como foi explicado inicialmente, os utilizadores estão sujeitos a aprovação por parte de um administrador. Este processo é controlado no ficheiro 'UserController.php', o seguinte código exemplifica o processo de validação de um novo utilizador:

```
1. public function ativar($id)
2. {
3.     $user = User::find($id);
4.     $user->status = 'ativo';
5.     $user->save();
6.     $user->notify(new NovoRegistoAtivado);
7.     $username = User::select('username')->where('id', '=', $id)-
        >value('username');
8.     Activity::log('User alterado para Ativo. Username: '.$username);
9.     return Redirect::route('painel')
10.    ->withSuccess('User Ativo com Sucesso!');
11. }
```

Podemos agora verificar este módulo na figura 22, onde o administrador pode fazer toda a gestão dos mesmos:

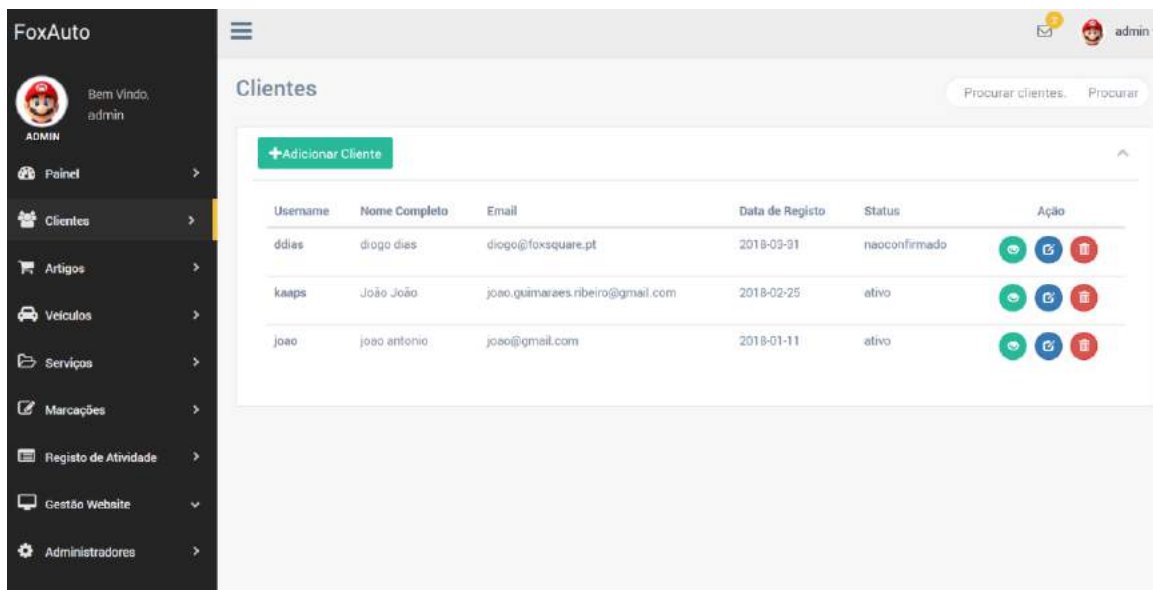


Figura 22 - Plataforma Web: Gestão de Utilizadores

De seguida o desenvolvimento recaiu sobre outro grande módulo deste projeto, gestão de automóveis:

Módulo Gestão de Automóveis

Tal como no módulo de gestão de utilizadores foi também desenvolvido para a gestão de automóveis um sistema de CRUD. Este módulo pode ser gerido por um administrador da plataforma ou um utilizador na sua própria área de gestão. Tem como propósito a possibilidade da própria oficina, adicionar novos carros atribuindo cada carro a um respetivo utilizador, ver, editar e apagar os mesmos. Neste mesmo módulo será incluído os registos dos automóveis com todas as funcionalidades da gestão de automóveis. Este irá focar-se mais em dados que estão em constante mudança no veículo. Estes processos são geridos em dois controladores, o ficheiro 'CarrosController.php' e 'RegistoCarrosController.php'.

Como se pode constatar na figura 23 é apresentada a listagem de todos os veículos dos utilizadores. É também possível ver uma breve descrição das características principais de cada veículo bem como o estado do serviço do mesmo.

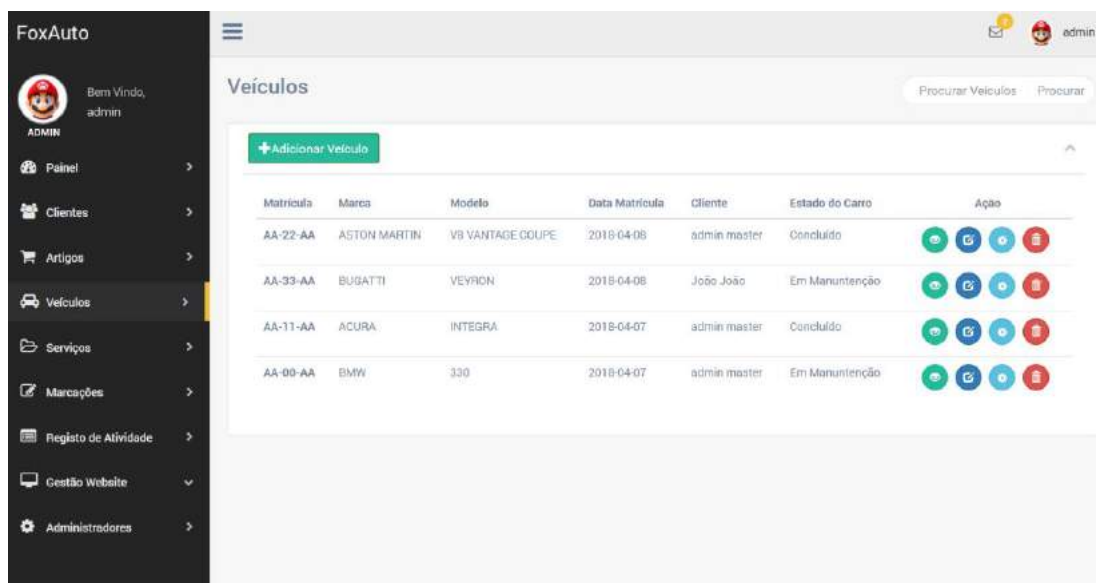


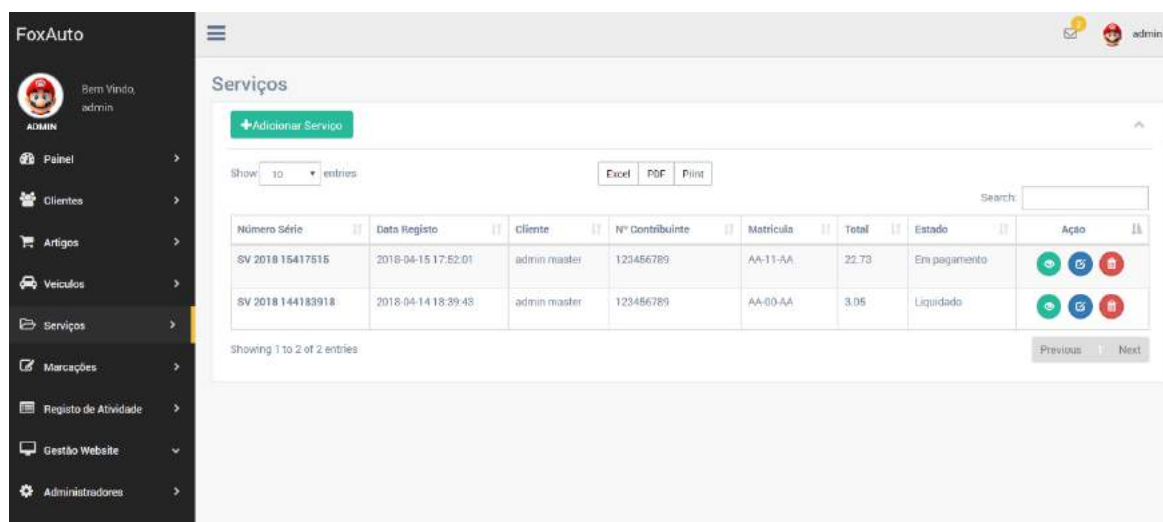
Figura 23 - Plataforma Web: Gestão de Automóveis

Depois da conclusão destes módulos, seguiu-se o módulo de gestão de serviços e artigos:

Módulo Gestão de Serviços e Artigos

O módulo de gestão de serviços tem como finalidade que a oficina possa registar as alterações que irá fazer aos automóveis dos clientes, incluindo possíveis descontos a determinados clientes. É também possível visualizar, editar e remover os mesmos (sistema CRUD). Este é complementado com o módulo de artigos onde é dada a possibilidade ao administrador da plataforma inserir artigos, o respetivo detalhe e preço unitário destes, que por sua vez também possui um sistema de *CRUD*. Estes dois módulos foram os que levaram mais tempo a serem concluídos, pois para além da inserção de campos de texto básicos é possível ver em tempo real uma 'fatura' com os artigos inseridos e devidos cálculos do total de custo, consoante o preço unitário, a quantidade, os descontos e o Iva. Este processo em tempo real foi possível recorrendo à tecnologia da linguagem de programação Vue.js. Este processo tem como principais ficheiros 'ArtigosController' e 'ServicosController'.

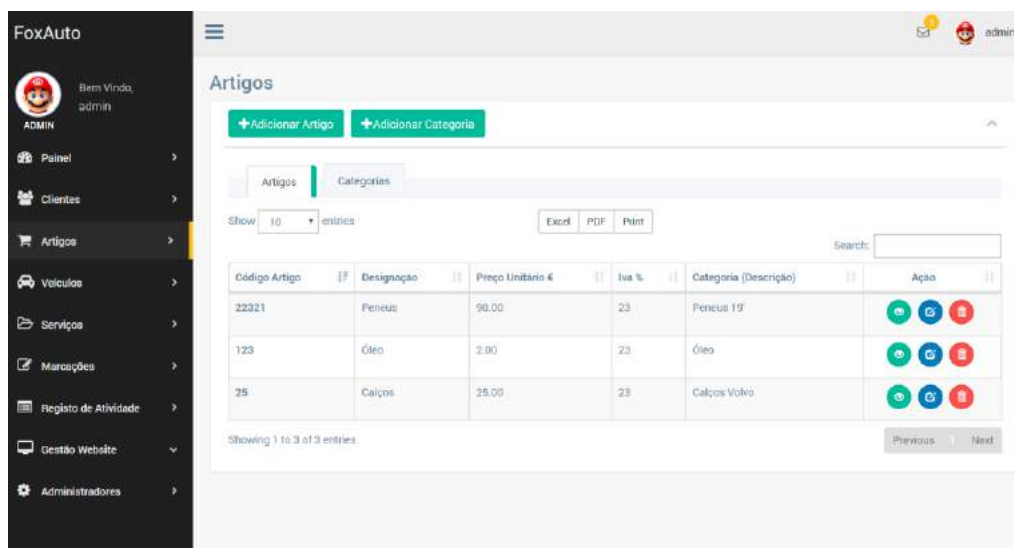
O administrador tem o privilégio de adicionar novos serviços realizados na oficina a um determinado veículo bem como todos os detalhes do mesmo, figura 24.



Número Série	Data Registo	Cliente	Nº Contribuinte	Matrícula	Total	Estado	Ação
SV 2018 15417515	2018-04-15 17:52:01	admin master	123456789	AA-11-AA	22.73	Em pagamento	[Edit] [Delete] [Add]
SV 2018 144183918	2018-04-14 18:39:43	admin master	123456789	AA-00-AA	3.95	Liquidado	[Edit] [Delete] [Add]

Figura 24 - Plataforma Web: Gestão de Serviços.

Como administrador tem o acesso à gestão de artigos como demonstra a figura 25.



Código Artigo	Designação	Preço Unitário €	Iva %	Categoria (Descrição)	Ação
22321	Pneus	90.00	23	Pneus 19"	[Edit] [Delete] [Add]
123	Óleo	2.00	23	Óleo	[Edit] [Delete] [Add]
25	Calços	25.00	23	Calços Volvo	[Edit] [Delete] [Add]

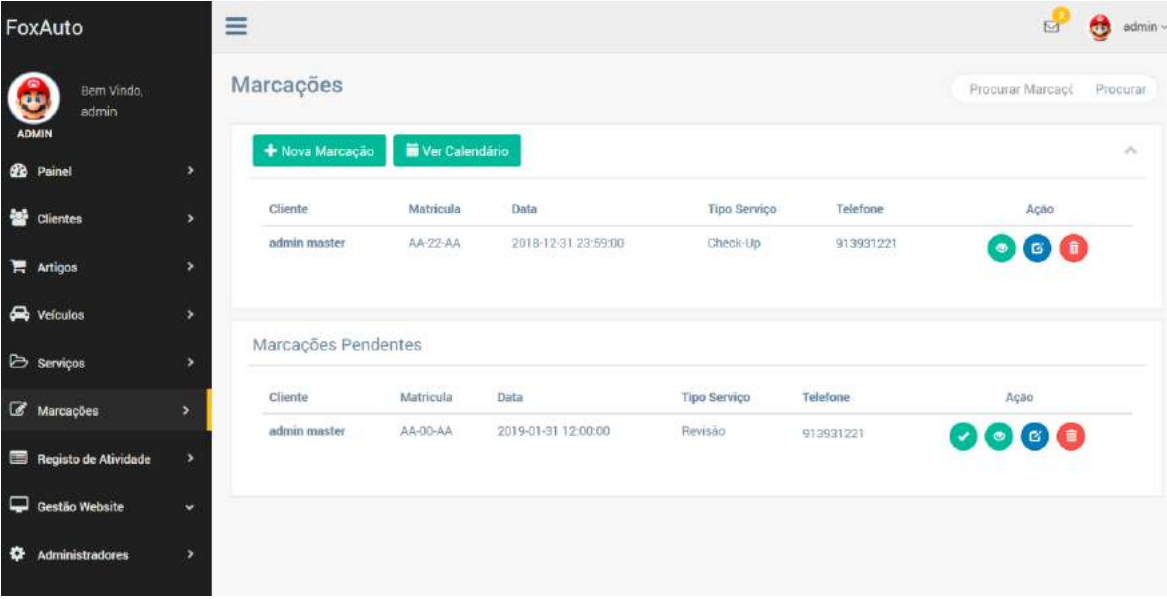
Figura 25 - Plataforma Web: Gestão de Artigos

Após os módulos de gestão de serviços e artigos estarem concluídos e a funcionar devidamente, passou-se ao desenvolvimento do módulo de gestão de marcações:

Módulo Gestão de Marcações

Tal como os módulos anteriores este módulo possui um sistema CRUD, e tem como ambição que o próprio utilizador possa fazer pedidos de marcações para o seu veículo, podendo escolher um automóvel entre os seus previamente adicionados à plataforma, selecionar o tipo de serviço, a data e hora. O administrador, por sua vez, poderá ver todos os pedidos de marcações, aceitar, editar e apagar os mesmos. Esta gestão é feita através do ficheiro 'MarcacoesController.php'.

Na figura 26 é apresentada a listagem de todas as marcações atuais, é possível ver as mesmas em forma de listagem ou calendário. Nesta página é possível criar uma nova marcação e ao administrador é possível gerir as marcações pendentes a pedido dos utilizadores da plataforma.



The screenshot shows the 'FoxAuto' web application interface. On the left is a dark sidebar with the 'FoxAuto' logo and a user profile for 'ADMIN'. The main content area is titled 'Marcações' and features a search bar and two buttons: '+ Nova Marcação' and 'Ver Calendário'. Below these are two tables. The first table, 'Marcações', lists a booking for 'admin master' with license 'AA-22-AA' on '2018-12-31 23:59:00' for a 'Check-Up' service at phone '913931221'. The second table, 'Marcações Pendentes', lists a pending booking for 'admin master' with license 'AA-00-AA' on '2019-01-31 12:00:00' for a 'Revisão' service at phone '913931221'. Both tables include action icons for edit, view, and delete.

Cliente	Matricula	Data	Tipo Serviço	Telefone	Ação
admin master	AA-22-AA	2018-12-31 23:59:00	Check-Up	913931221	[Edit] [View] [Delete]

Cliente	Matricula	Data	Tipo Serviço	Telefone	Ação
admin master	AA-00-AA	2019-01-31 12:00:00	Revisão	913931221	[Check] [Edit] [View] [Delete]

Figura 26 - Plataforma Web: Gestão de Marcações

De seguida no desenvolvimento deste projeto foi desenvolvido o módulo de estatísticas:

Módulo Estatísticas

Este módulo tem como objetivo projetar ao administrador de uma forma eficaz e rápida uma visão global da plataforma no painel principal, através das várias informações desta, dando algumas estatísticas rápidas como a quantidade de utilizadores registados, últimos utilizadores registados, total de clientes registados na plataforma entre outras estatísticas. A gestão deste módulo é maioritariamente feita no ficheiro 'PainelController.php'.

Nesta figura 27 podemos ver o painel de Administrador, é nesta página da plataforma que podemos ver algumas destas estatísticas.

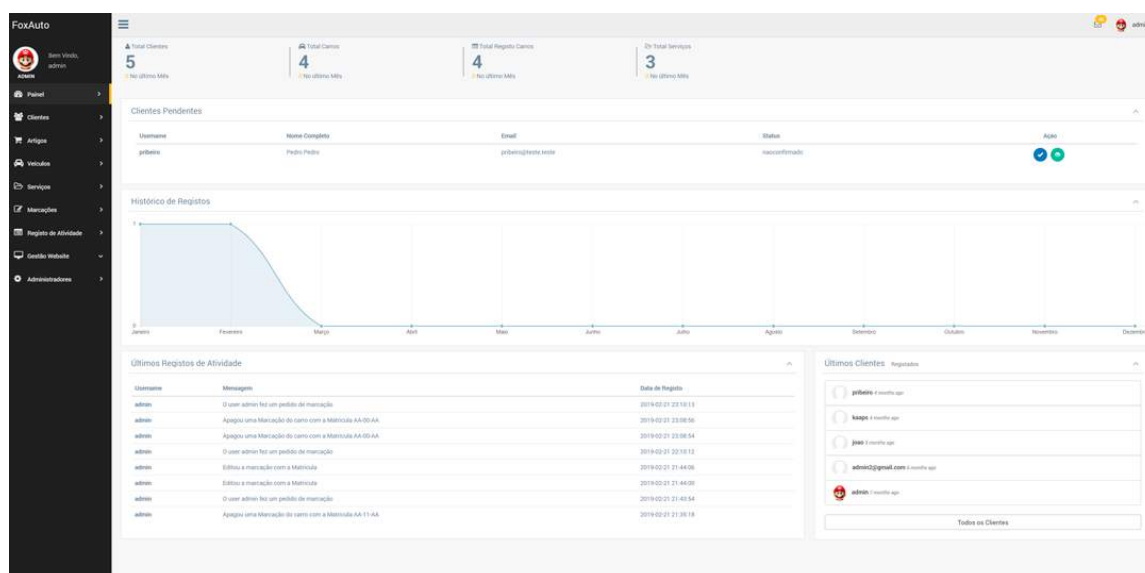


Figura 27 - Plataforma Web: Painel

Já como uma boa base na plataforma foi desenvolvido o módulo de registo de atividade:

Módulo de Registo de Atividade

Para que o administrador possa continuamente ter o fácil acesso a todas as atividades da plataforma, por exemplo, quando alguém altera o seu email ou até mesmo quando um novo veículo é inserido no sistema, decidiu-se desenvolver este módulo à qual apenas o administrador tem acesso. O mesmo é gerido no ficheiro 'ActivityController'.

Na seguinte figura 28, pode-se verificar os vários diferentes tipos de registo de atividade:

Username	Mensagem	IP	Data de Registo	Ação
admin	Editou a marcação com a Matrícula	192.168.1.1	2019-01-19 22:51:34	
admin	Editou a marcação com a Matrícula	192.168.1.1	2019-01-19 22:51:15	
admin	Adicionou artigo com o Código Artigo 25	192.168.1.1	2019-01-19 22:27:01	
admin	Adicionou Categoria com a Descrição Calços Volvo	192.168.1.1	2019-01-19 22:26:43	
admin	Apagou um Artigo com o Código Artigo 1	192.168.1.1	2019-01-19 22:26:13	
admin	Editou Artigo com o cod_artigo 123	192.168.1.1	2019-01-19 22:26:05	
admin	Editou Artigo com o cod_artigo 1	192.168.1.1	2019-01-19 22:25:47	
admin	Editou Artigo com o cod_artigo 22321	192.168.1.1	2019-01-19 22:25:14	
admin	Editou Artigo com o cod_artigo 1	192.168.1.1	2019-01-19 22:24:46	
admin	Editou Artigo com o cod_artigo 123	192.168.1.1	2019-01-19 22:24:40	

Figura 28 - Plataforma Web: Registo de Atividade

Após a conclusão do módulo anterior sentiu-se a necessidade de existir mais de um administrador da plataforma. Neste sentido, foi desenvolvido o módulo de gestão de administradores:

Módulo Gestão Administradores

Este módulo, tal como o nome indica, apenas é acessível por parte de membros da administração. O mesmo tem um sistema 'CRUD', onde o principal objetivo é a inserção de novos administradores na plataforma e a gestão destes. Este processo é realizado no ficheiro 'AdministradoresController.php'.

Pode-se verificar agora na figura 29 esta Gestão de Administradores:

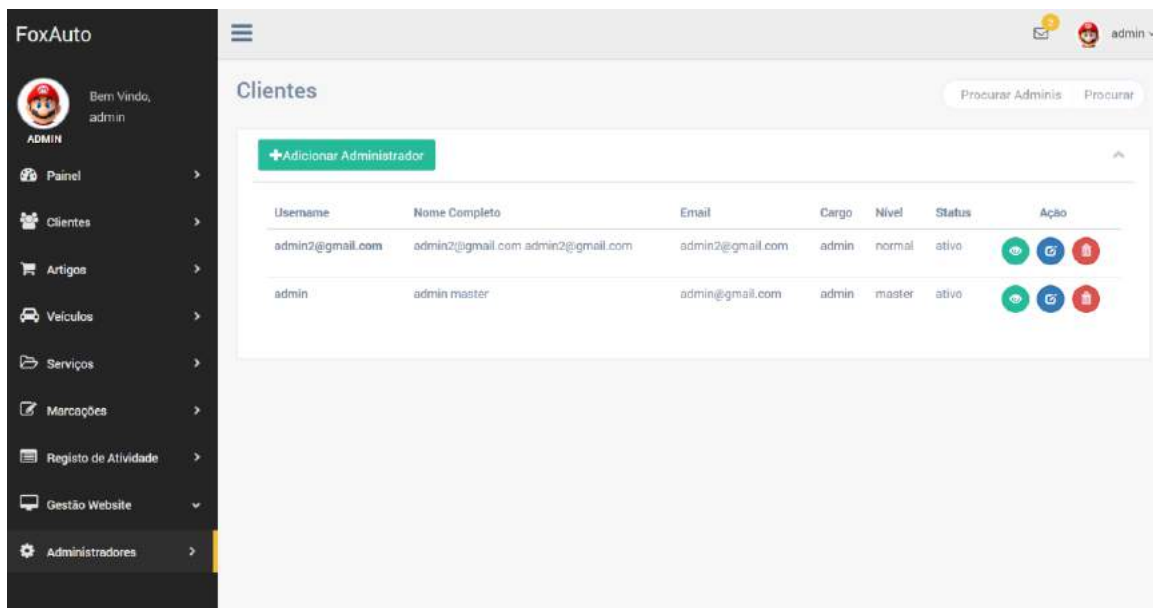


Figura 29 - Plataforma Web: Gestão de Administradores

Com o término do módulo de gestão de administradores seguiu-se o desenvolvimento da *API*:

Módulo API

Como abordado anteriormente neste documento, este módulo foi desenvolvido recorrendo à biblioteca *JWT-Auth*. A escolha recaiu sobre esta, pois é de código aberto e transmite de forma compacta e segura as informações entre as partes em objetos *JSON*. O facto de os dados serem transmitidos em formato *JSON*, contém várias valias à transmissão dos seus, principalmente entre a plataforma web e a aplicação móvel. Por sua vez, estes são codificados permitindo a redução do espaço ocupado. Iniciou-se pela inclusão do sistema de autenticação do *JWT* no sistema de login do Laravel, pois quando o utilizador estiver autenticado cada solicitação à *API* subsequente incluirá o *JWT*, permitindo que este ascenda às *routes*, serviços e recursos permitidos com o *token* associado. O seguinte código mostra como é feita esta nova autenticação e atribuído o *token*:

```

1. $token = JWTAuth::fromUser($user);
2. $objectToken = JWTAuth::setToken($token);
3. $expiration = JWTAuth::decode($objectToken->getToken())->get('exp');
4. $data = array(
5.     'access_token' => $token,
6.     'token_type' => 'bearer',
7.     'expires_in' => \Config::get('jwt.ttl') * 60
8. );
9. return response()->json(['userData' => $data]);

```

O próximo passo foi definir todas as *routes* necessárias da *API*, para que a aplicação móvel possa fazer um pedido à *API* e esta consiga navegar na plataforma web e devolver os dados pretendidos. Alguns exemplos destas *routes*:

```

1. Route::group(['prefix' => 'api'], function()
2. {
3.     Route::get('/', function () {
4.         return response()-
5.             >json(['message' => 'Fox API', 'status' => 'Connected']);
6.     });
7.     Route::post('getauthuser', 'ApiController@getAuthUser');
8.     Route::post('createUser', 'Auth\RegisterController@createUser');
9.     Route::post('auth/login', 'AuthController@authenticate');
10. });

```

Estando as *routes* criadas, passou-se ao desenvolvimento do código necessário, no controlador 'ApiController.php' para que seja permitida a devolução dos dados aquando o pedido. Como anteriormente foi atribuído o *token*, no início de cada função é necessária a verificação deste, caso exista o *token*, os dados são devolvidos em JSON tal como pretendido. Pode-se agora ver, neste código, a verificação desse *token* e a devolução dos dados pretendidos; neste caso todos os automóveis do utilizador autenticado:

```

1. public function getAuthUserCarro(Request $request){
2.     $user = JWTAuth::toUser($request->token);
3.     $carros = DB::table('carros')
4.     ->join('registo_carro', 'carros.id', '=', 'registo_carro.id_carro')
5.     ->select('carros.*', 'registo_carro.*')
6.     ->where('carros.id_user', '=', $user->id)
7.     -
8.     >whereRaw('registo_carro.id IN ( SELECT MAX(id) FROM registo_carro GROUP BY regis
9.     to_carro.id_carro )')
10.     ->get();
11.     return response()->json(['userCarros' => $carros]);
12. }

```

Na figura 30 foi feito um pedido à *API* para a autenticação do utilizador admin:

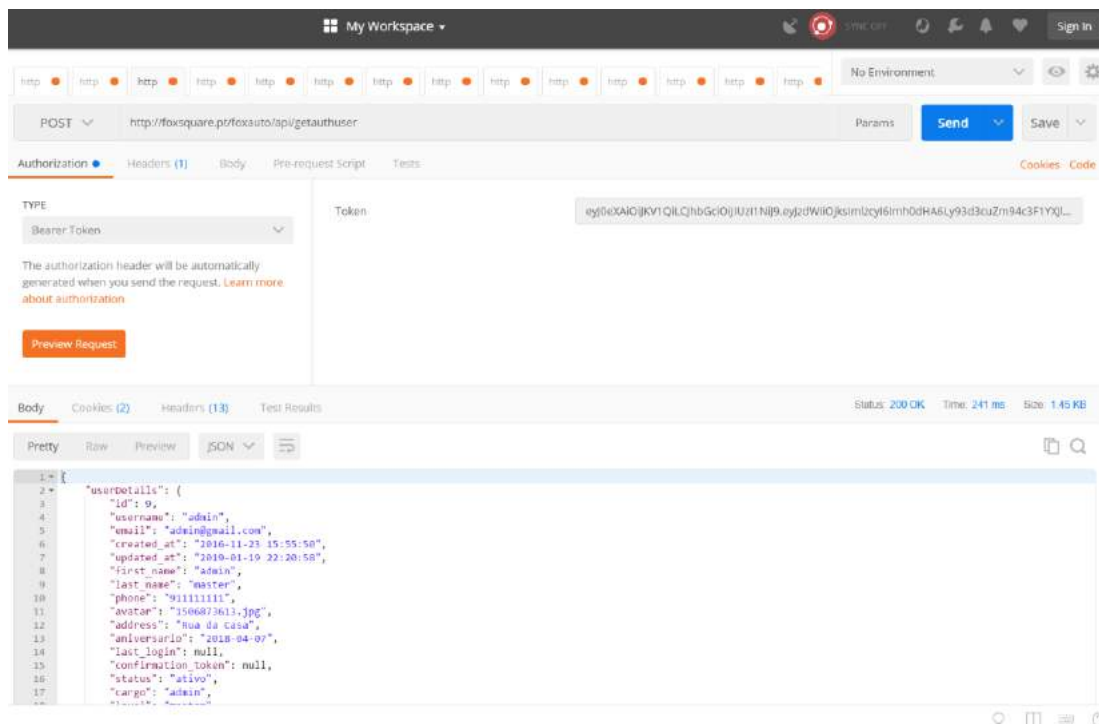


Figura 30 - Plataforma Web: API

Após o desenvolvimento de todas as funções necessárias para o bom funcionamento da *API*, seguiu-se a elaboração do módulo de páginas informativas:

Módulo Páginas Informativas

Tem como objetivo dar a conhecer a oficina/stand a qualquer pessoa que aceda ao website, aceder às várias páginas informativas, entre as quais serviços, contactos e automóveis para venda. Os automóveis para venda são diferentes dos automóveis já descritos anteriormente, pois estes são automóveis que a oficina/stand tenha disponíveis para comercialização. Para tal foram criadas as tabelas *automoveis*, *automoveis_marca*, *automoveis_extra*, *automoveis_modelo* e *automoveis_fotos* na base de dados. Estas páginas têm o suporte dos controladores 'FrontEndController.php' e 'AutomoveisController.php' e as respetivas *views*.

Na figura 31 é apresentado a página inicial da plataforma web onde o utilizador tem acesso rápido a todas as zonas do website, permitindo também fazer uma pesquisa detalhada pelo veículo que procura.



Bem Vindo à FoxAuto

O cliente é a razão da nossa empresa. A satisfação das suas necessidades é o nosso objectivo.

Figura 31 - Plataforma Web: Home

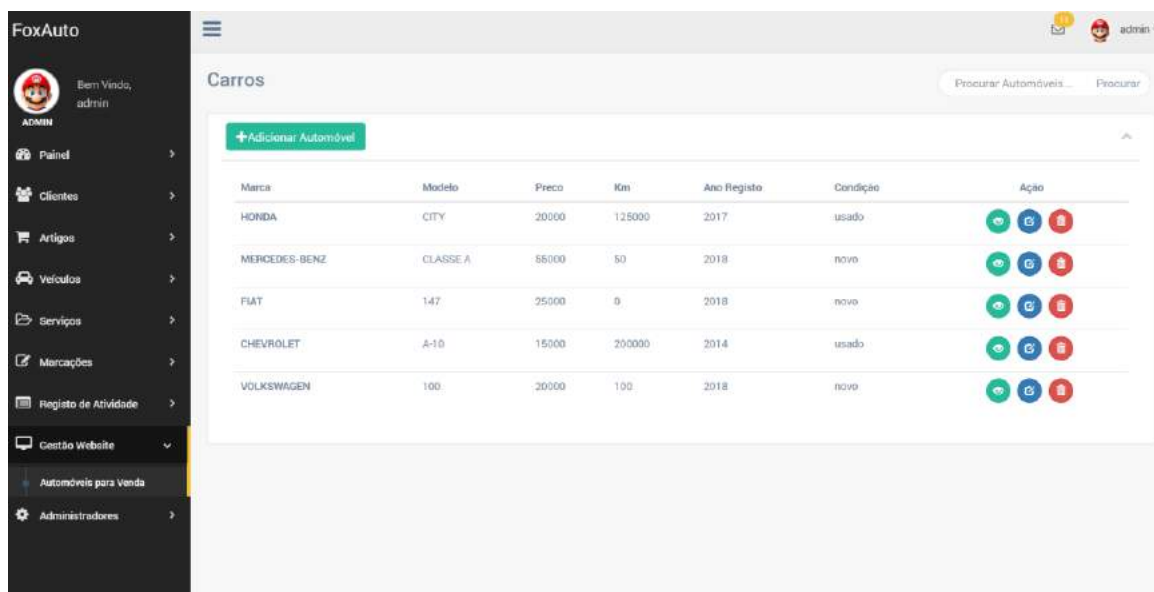
Com o desenvolvimento do módulo anterior sentiu-se a necessidade de criar um módulo onde fosse possível gerir estes automóveis para venda. Assim, criou-se o módulo de gestão de automóveis para venda:

Módulo Automóveis para Venda

Tem como propósito que a oficina possa adicionar, de forma rápida e eficaz, carros que tenha para venda no seu stand e que estes possam aparecer nas páginas do website. Tal como nos módulos anteriores, foi criado um sistema

‘CRUD’ que cria este suporte de inserção, leitura, edição e remoção destes dados. Esta gestão é feita na diretoria ‘GestaoWebsiteController.php’.

Como se pode verificar na figura 32, o administrador pode gerir facilmente todos os automóveis para venda que estão inseridos na plataforma:


















Marca	Modelo	Preço	Km	Ano Registo	Condição	Ação
HONDA	CITY	20000	125000	2017	usado	  
MERCEDES-BENZ	CLASSE A	55000	50	2018	novo	  
FIAT	147	25000	0	2018	novo	  
CHEVROLET	A-10	15000	200000	2014	usado	  
VOLKSWAGEN	100	20000	100	2018	novo	  

Figura 32 - Plataforma Web: Gestão de Automóveis para Venda

Com todos estes módulos desenvolvidos e a funcionar devidamente passou-se ao desenvolvimento do último módulo:

Módulo Mensagens

Este módulo tem como simples função a visualização de mensagens que a plataforma vai enviando automaticamente aos utilizadores, ou seja, quando um automóvel está pronto para ser entregue ao cliente, para além deste receber um email, recebe também, uma mensagem na plataforma com esta informação. O administrador pode também, caso deseje, enviar uma mensagem específica a um determinado utilizador da plataforma.



Figura 33 - Plataforma Web: Mensagens

Como se pode verificar na figura 33, um dos possíveis administradores da plataforma pode facilmente ver as mensagens enviadas, as mensagens pessoais, as arquivadas, ou então, enviar uma nova mensagem.

4.4 Desenvolvimento e Implementação da Aplicação Móvel

Pretende-se agora dar a conhecer todos os processos e fases de desenvolvimento da aplicação móvel, começando pela estrutura desta e em seguida apresentar todos os módulos desenvolvidos que levaram à realização desta aplicação.

4.4.1 Estrutura

É apresentada na figura 34 a estrutura da aplicação móvel, esta estrutura é praticamente igual para qualquer projeto desenvolvido com a *framework IONIC*, pois muitos dos ficheiros e pastas são gerados automaticamente ao criar um projeto novo:

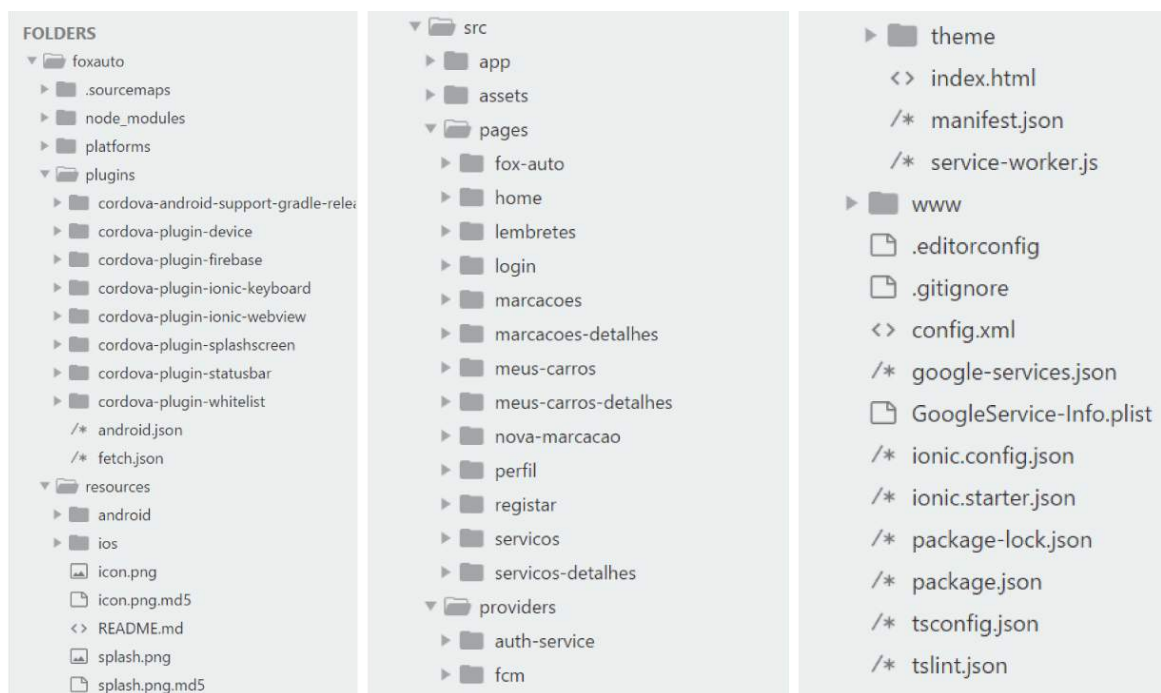


Figura 34 - Estrutura do Código da Aplicação Mobile

Será agora descrita algumas das pastas e ficheiros mais relevantes desta aplicação:

App: Contém todos os ficheiros necessários para a inicialização da aplicação e a principal estrutura onde será contruída maioritariamente a aplicação.

Pages: Esta pasta contém todas as páginas *HTML*, *JS* e *CSS* da aplicação.

App.component.ts: Este ficheiro define a estrutura básica e a navegação inicial da aplicação.

App.module.ts: Local onde se encontra o ponto inicial da aplicação. Este ficheiro inclui o módulo principal do *Angular 2 (NgModule)* da aplicação; é também onde são declaradas a maior parte das dependências.

Plugins: Pasta que contém todos os plugins externos utilizados pela aplicação.

Auth-service.ts: Ficheiro principal que faz a ligação entre a aplicação móvel e plataforma web; esta é feita através de pedidos realizados à *API* contruída anteriormente na plataforma web.

Variables.scss: Neste ficheiro encontram-se os estilos principais da aplicação.

4.4.2 Módulos da Aplicação Móvel

Nesta parte do documento serão explicados os processos realizados que permitiram o desenvolvimento desta aplicação. O propósito desta aplicação é fazer com que o prestador de serviços(oficina/stand) e o cliente pudessem de certa forma ter um contacto mais próximo entre os dois. Para tal a aplicação foi desenvolvida a pensar nisso, criou-se uma aplicação simples, de fácil uso e visualmente apelativa ao utilizador. Tal como na aplicação web, foi intenção sair da zona de conforto e abordar uma *framework* completamente nova, neste caso o *IONIC*. Este possui uma vasta documentação online e comunidade ativa. O seu desenvolvimento foi um pouco mais lento do que a aplicação web, uma vez que a linguagem base desta *framework* é o *Angular* houve uma maior e mais lenta curva de aprendizagem. Após exaustiva leitura da documentação e visualização de tutoriais online, deu-se início à instalação da *framework* e desenvolvimento de alguns processos simples como a aplicação do layout, a navegação entre as diversas páginas, etc. De seguida foi desenvolvido o módulo de ligação da aplicação móvel à *API* e em simultâneo o módulo de login e registo:

Módulo Ligação com a Api, Login e Registo

Este módulo é o mais importante de toda a aplicação mobile pois faz com que seja possível que os dados do sistema web sejam transmitidos para a aplicação mobile e vice-versa. Tal como explicado anteriormente, para ser possível a troca destes dados, é necessária uma *token*, esta é gerada quando é feito o login por um utilizador. Sabendo isto é necessário que haja um pedido à *API* por parte da aplicação mobile aquando o login e que este verifique se é possível fazer a autenticação, devolvendo os dados do utilizador e a devida *token*. Este pedido é feito em dois ficheiros 'login.ts' e 'auth-service.ts'. Poder-se-á verificar agora o código necessário para a obtenção destes dados:

Login.ts

```
1. goToFoxAuto(params){
2.   this.authServiceProvider.postData(this.userData, "auth/login").then((result
   ) =>{
3.     this.responseData = result;
4.     localStorage.setItem('userData', JSON.stringify(this.responseData));
5.     this.navCtrl.push(FoxAutoPage);
6.   }, (err) => {
7.     console.log(err);
8.     this.presentAlert(err.error);
9.   });
10. }
```

Auth-service.ts

```
1. let apiUrl = "http://192.168.1.78:8000/api/";
2. postData(credentials, type) {
3.   this.showLoading();
4.   return new Promise((resolve, reject) => {
5.     let headers = new HttpHeaders({ 'Content-Type': 'application/json' });
6.
7.     this.http.post(apiUrl + type, JSON.stringify(credentials), {headers: head
   ers})
8.       .subscribe(res => {
9.         resolve(res);
10.      }, (err) => {
11.        reject(err);
12.      });
13.     this.dismissLoading();
14.   });
15. }
```

Como se pode constatar, é feito um pedido na página de login ao *provider* (auth-service) e este, por sua vez, faz o pedido à *API*. Caso seja executado com sucesso, os dados são guardados localmente enquanto o tempo da *token* não expirar. Nesta altura, o utilizador está autenticado e a aplicação tem acesso à *token* e aos dados deste utilizador.

Para que a aplicação possa ter acesso aos restantes dados, como marcações, automóveis, etc., é necessário haver novos pedidos: desta vez apenas a *token* é necessária ser enviada. Estes pedidos são também realizados em vários ficheiros consoante os dados pretendidos, sendo que todos estes dados têm que passar pelo ficheiro 'auth-service.ts':

```

1. getDataToken(credentials, type) {
2.   this.showLoading();
3.   return new Promise((resolve, reject) => {
4.     let headers = new HttpHeaders({ 'Content-
      Type': 'application/json', 'Accept': 'application/json', 'Authorization': 'Bearer
      ' + credentials.token });
5.     this.http.get(apiUrl + type, {headers: headers})
6.       .subscribe(res => {
7.         resolve(res);
8.       }, (err) => {
9.         reject(err);
10.      });
11.     this.dismissLoading();
12.   });
13. }

```

Na figura 35 é apresentada a página final de login, onde o utilizador necessita de inserir o email e a password para ter acesso, ou então, fazer um novo registo e esperar que este seja aceite por um administrador.

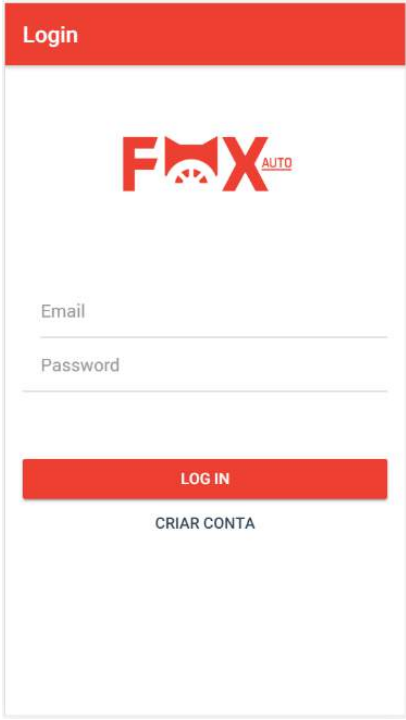


Figura 35 - Aplicação Mobile: Login

Após a conclusão destes módulos seguiu-se o módulo de informações de utilizador (Perfil):

Módulo Informações do Utilizador

Este módulo tem como função apenas mostrar os dados principais do utilizador que está autenticado. Este foi mais simples de criar visto que anteriormente já tinham sido criados todos os elementos necessários, para que fosse possível fazer os pedidos de uma forma simples à *API*.

Na figura 36 mostra como estas informações são visualizadas pelo utilizador na aplicação:

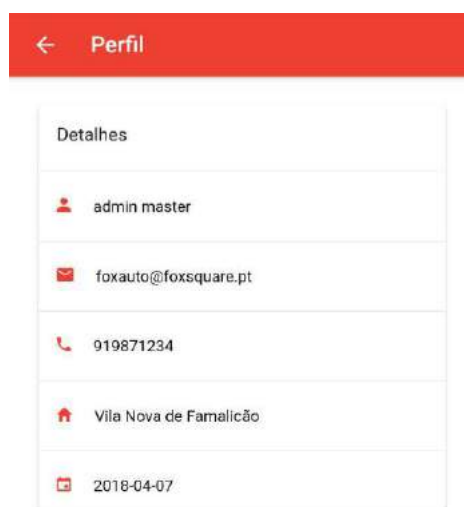


Figura 36 - Aplicação Mobile: Perfil

Com o módulo concluído seguiu-se o desenvolvimento do módulo automóveis do utilizador:

Módulo Automóveis do utilizador

Tem como propósito mostrar todos os dados associados às viaturas de um determinado utilizador. Para o efeito, é necessário fazer um novo pedido à *API*, para que esta retorne todos os dados e registos dos automóveis do utilizador.

Na imagem apresentada (figura 37) pode-se observar a página relativa aos 'Meus Carros'. Esta apresenta o estado do carro, marca e modelo. Se o utilizador fizer clique num destes automóveis, poderá visualizar noutra página, mais detalhes desse veículo.



Figura 37 - Aplicação Mobile: Meus Carros

Com mais um módulo concluído, segue-se o desenvolvimento do módulo de serviços:

Módulo Serviços

Neste módulo, tal como no módulo apresentado anteriormente, o objetivo é apenas a possibilidade da visualização de dados dos vários serviços realizados a

um automóvel do utilizador. Para tal, foi criada uma *list-view* permitindo a fácil visualização destes dados. Estes dados são recebidos na aplicação após o pedido feito à *API*.

No módulo de serviços na aplicação móvel, é apresentado inicialmente o número de série do serviço e o estado do mesmo. Para ver mais detalhes, basta clicar num dos serviços e a aplicação abrirá uma nova página com os mesmos:



Figura 38 - Aplicação Mobile: Serviços

Após a conclusão de mais um módulo da aplicação mobile seguiu-se o desenvolvimento do módulo de marcações da mesma:

Módulo Marcações

O Módulo de marcações na aplicação mobile permite ao utilizador visualizar todas as manutenções agendadas na oficina e verificar o estado da marcação, ou seja, quando uma marcação é feita fica pendente de aprovação por parte de um membro da administração. Este módulo permite também o utilizador efetuar um pedido de marcação. Assim sendo, o utilizador escolhe a matrícula do carro em questão, o tipo de serviço, a data e hora e, se necessário, observações.

Posteriormente e, caso a marcação seja aceite, o cliente recebe um email a notificar que foi aceite e as devidas informações.

Na figura 39 pode-se verificar como é feito um novo pedido de marcação:

A imagem mostra a interface de uma aplicação móvel para criar uma nova marcação. O formulário tem um cabeçalho vermelho com o texto "Nova Marcação" e um ícone de seta para trás. Os campos do formulário são:

- Matrícula: AA-00-AA
- Serviço: Revisão
- Data e Hora: 2019/01/17 11:00
- Observações: Revisão Geral

Na base do formulário, há um botão vermelho com o texto "NOVA MARCAÇÃO".

Figura 39 - Aplicação Mobile: Nova Marcação

Com a conclusão do módulo de marcações passou-se ao desenvolvimento do módulo de notificações.

Módulo Lembretes/Notificações

O módulo de lembretes/notificações foi concebido de modo a facilitar a interação entre o utilizador e a oficina, ou seja, desenvolveu-se de forma a que estes sejam enviados automaticamente pelo sistema. Tem como objetivo retirar a preocupação ao cliente, nomeadamente em situações como: levar o carro à inspeção/revisão; buscar o carro à oficina depois da sua manutenção e lembrar a data do pagamento do IUC ou do seguro do automóvel. O desenvolvimento destas notificações foi possível recorrendo a dois sistemas: o *Google Cloud*

Messaging, onde é feito o registo das credenciais e chaves da aplicação, e ao *OneSignal* serviço de notificações *Push*. Começou-se então por registar a aplicação no *Google Cloud Messaging*. Com este registo foi possível obter as credenciais e chaves da aplicação; de seguida foi instalado na *Framework IONIC* o plugin do *OneSignal*. Depois de instalado, inseriu-se as credenciais e chaves obtidas anteriormente e fez-se as configurações e programação necessárias para que ambas funcionassem corretamente. Por fim, e visto que anteriormente desenvolveu-se o módulo de emails automáticos, foi possível adicionar o envio das notificações para a aplicação móvel sempre que, por exemplo, um email de aviso de inspeção é enviado. Para que tudo isto funcione de forma autónoma e sem qualquer intervenção de um administrador da plataforma, foi adicionado um *cronjob* ao serviço de alojamento web para que todos os dias leia estes ficheiros e assim, caso exista alguma notificação a ser enviada, este envia sozinho. Poder-se-á agora verificar, na figura 40 no ecrã da esquerda, que um determinado utilizador recebeu uma notificação a informar que o seu carro já se encontra pronto e, no ecrã da direita, poder-se-á verificar todos os lembretes associados a uma determinada matrícula de um utilizador, desde que aconteçam durante o mês corrente, excetuando o lembrete da inspeção, o qual é notificado com um mês de antecedência:

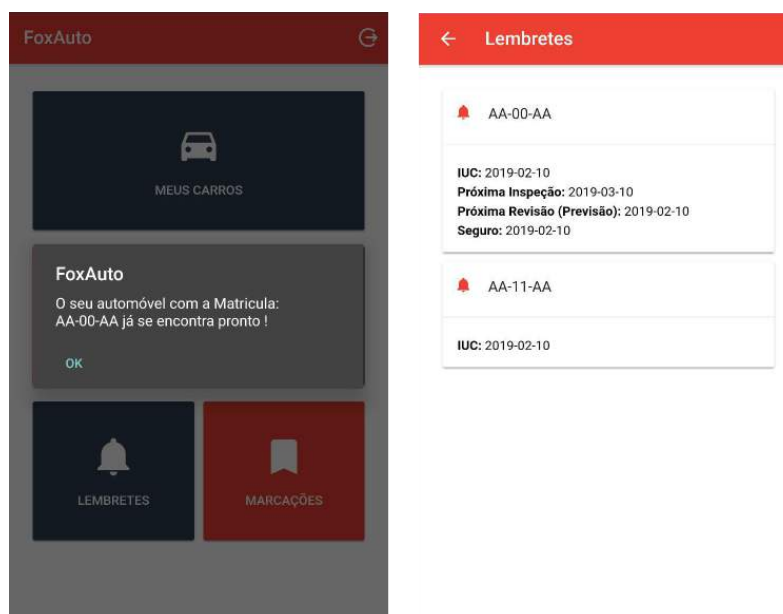


Figura 40 - Aplicação Mobile: Notificação e Lembretes

Na figura 41 pode-se verificar como é realizada a interação entre o dispositivo móvel, a plataforma de gestão e o serviço de notificações *Push*:

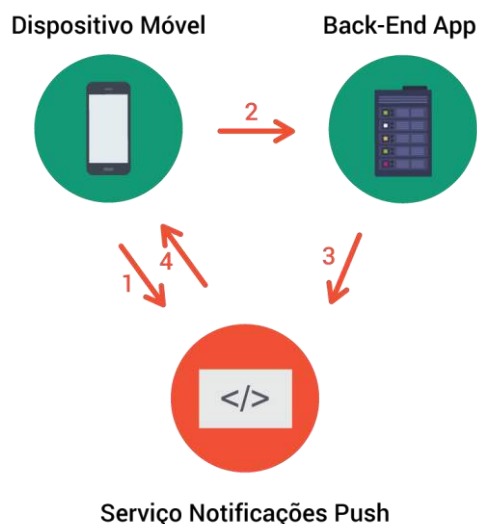


Figura 41 - Esquema Notificações

1. O dispositivo é registado num serviço de Notificações Push (OneSignal);
2. O dispositivo adquire uma token;
3. O servidor envia um texto de notificação para o serviço de notificações push;
4. O serviço de notificações push encaminha o texto de notificação para o dispositivo.

4.6 Manutenção das Plataformas de Desenvolvimento

Foi escolhida a framework *Laravel* de modo a manter a boa funcionalidade do sistema, pois a mesma está estruturada com a design *pattern Model View Controller (MVC)*, o que facilita que qualquer programador consiga facilmente entender a estrutura da mesma, podendo assim, ajustar e alterar a mesma caso necessário. Por sua vez, esta framework é bastante utilizada, pois possui uma vasta gama de documentação online e constantes atualizações. A componente

mobile foi desenvolvida em IONIC. Esta é também uma framework que tem como base o *AngularJs*. Possui ainda, muita documentação online e se for necessário, qualquer programador poderá atualizar a mesma.

5. TESTES E ANÁLISE DE RESULTADOS

Este capítulo apresenta as metodologias para testar a plataforma web e aplicação móvel. Serão analisadas a nível de testes de usabilidade, experiência do utilizador e validação de requisitos

5.1 Testes de Usabilidade

Os testes de usabilidade são utilizados para determinar o grau da qualidade da aplicação em termos de usabilidade e medir o quão fácil ou difícil estes são de utilizar. Sendo assim, irá ser utilizada a metodologia *System Usability Scale* (*SUS*). Esta consiste num questionário de escala simples, com 10 questões que oferece uma visão global de avaliações de usabilidade (Brooke, 1995). As mesmas são respondidas numa escala de 1 (discordo fortemente) a 5 (concordo fortemente). Serão agora apresentadas estas dez questões (Martins, Rosa, Queirós, Silva, & Rocha, 2015):

1. Acho que gostaria de utilizar este produto com frequência;
2. Considerei o produto mais complexo do que necessário;
3. Achei o produto fácil de utilizar;
4. Acho que necessitaria de ajuda de um técnico para conseguir utilizar este produto;
5. Considerei que as várias funcionalidades deste produto estavam bem integradas;
6. Achei que este produto tinha muitas inconsistências;
7. Suponho que a maioria das pessoas aprenderia a utilizar rapidamente este produto;
8. Considerei o produto muito complicado de utilizar;
9. Senti-me muito confiante a utilizar este produto;
10. Tive que aprender muito antes de conseguir lidar com este produto.

Benefícios da utilização do SUS

Qualquer pessoa consegue entender como os utilizadores sentem-se em relação à usabilidade das aplicações, apenas analisando um único número (Bernazzani, 2018).

O SUS é fácil de utilizar, possui uma escala de avaliação e é fácil enviar ao utilizador final, através de email ou outro método qualquer (Bernazzani, 2018).

Interpretar as pontuações

De acordo com o website usability.gov e a publicação no *Journal of Usability Studies* (Brooke, 2013), *interpretar a pontuação pode ser complexo. As pontuações dos participantes para cada pergunta são convertidas num novo número, somadas e multiplicadas por 2.5 para converter as pontuações originais de 0 a 40 para 0 a 100, estas não são percentagens e devem ser consideradas apenas em termos do seu ranking percentual. Com base em pesquisas, uma pontuação SUS acima de 68 será considerada acima da média e qualquer resultado abaixo de 68 estará abaixo da média* (Usability.gov, 2013).

Interpretação dos resultados

Com a recolha e análise dos dados, obteve-se um resultado de 91,81 pontos na aplicação móvel na qual foram inquiridas 11 pessoas das quais, 72,7% encontram-se na faixa etária compreendidas entre 24 e 30 anos; 18,2% entre os 19 e 23 anos e 9,1% entre os 31 e 40 anos. Na plataforma web o resultado de 95,71 pontos num total de 7 pessoas das quais, 71,4% encontram-se na faixa etária compreendidas entre 24 e 30 anos; 14,3% entre os 19 e 23 anos e 14,3% entre os 31 e 40 anos.

Segundo o sistema de usabilidade (SUS), a pontuação acima de 68 é considerada acima da média. Uma vez que ambos os sistemas obtiveram um resultado acima da média, pode-se concluir que estes cumprem os requisitos de uma boa aplicação a nível de usabilidade.

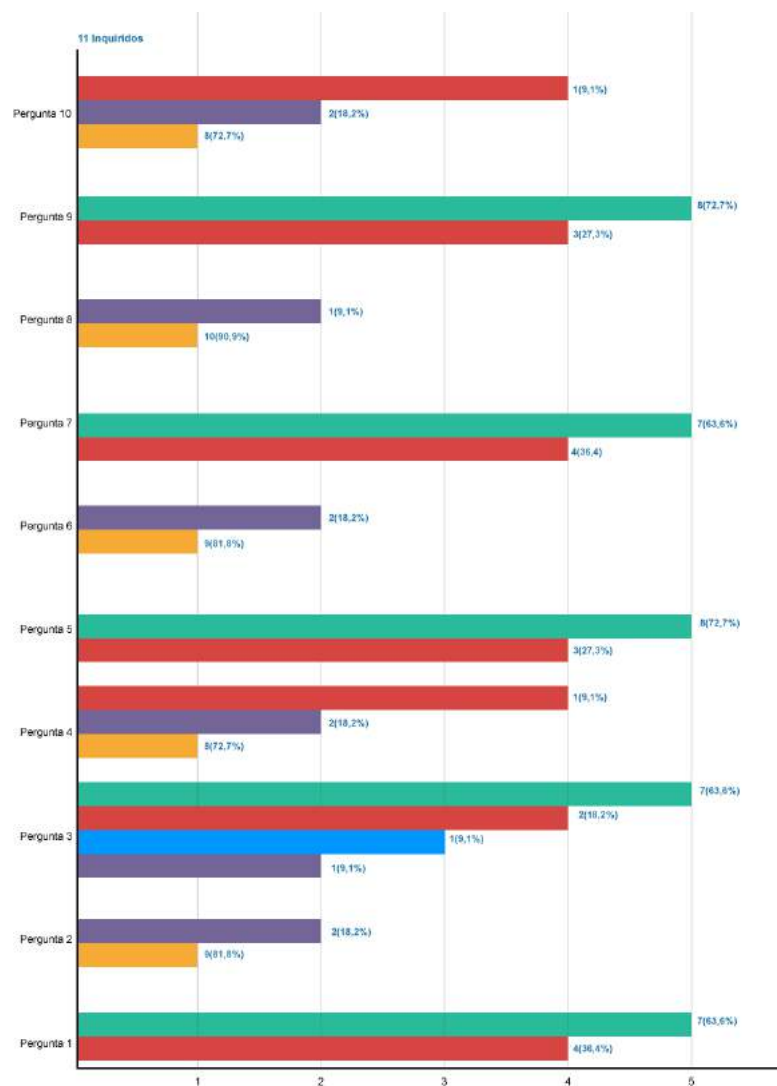


Figura 42 - Gráfico de Resultados SUS

Na figura acima representada, pode-se verificar o resultado obtido às 10 perguntas do SUS para a aplicação móvel. Estas questões foram respondidas numa escala de 1 (discordo fortemente) a 5 (concordo fortemente).

Destaca-se agora o resultado de algumas perguntas às quais é importante realçar a avaliação feita pelos inquiridos:

Acho que gostaria de utilizar este produto com frequência

11 respostas

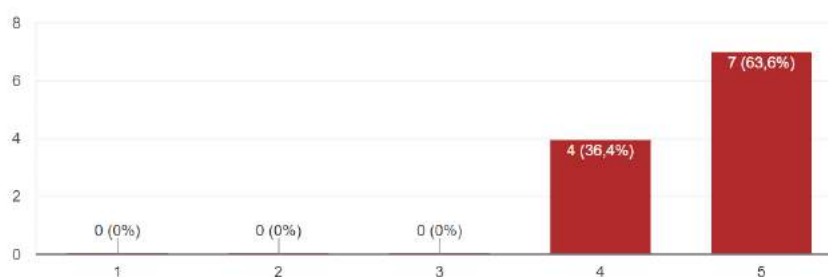


Figura 43 - Resultado SUS Pergunta 1

Com o resultado obtido positivamente à primeira questão, reforça-se a importância do desenvolvimento desta dissertação, pois disponibiliza de forma simples os dados dos automóveis aos utilizadores, mas também, a aproximação do cliente com a oficina. Resultado à quinta pergunta do questionário SUS:

Considerarei que as várias funcionalidades deste produto estavam bem integradas

11 respostas

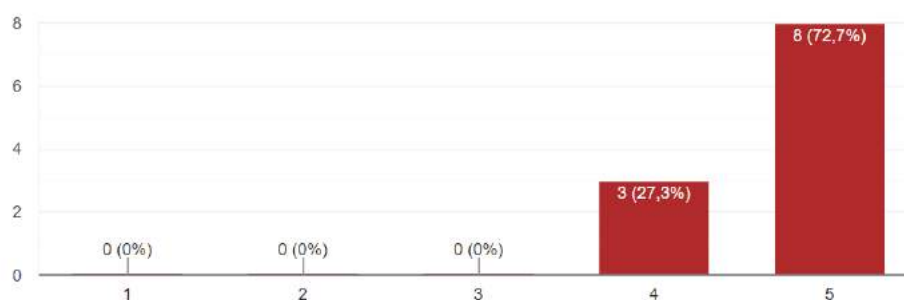


Figura 44 - Resultado SUS Pergunta 5

Pode-se verificar com o resultado a esta questão que as várias funcionalidades da aplicação móvel foram bem definidas e executadas.

Achei que este produto tinha muitas inconsistências

11 respostas

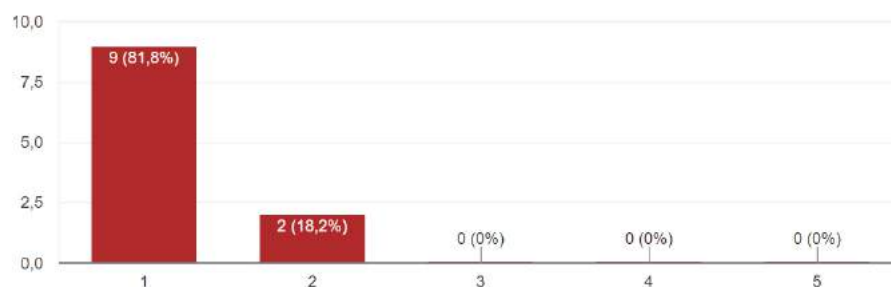


Figura 45 - Resultado SUS Pergunta 6

Na sexta pergunta do questionário SUS, pode-se averiguar com os resultados obtidos que a aplicação desenvolvida é toda ela consistente nas diferentes funcionalidades e objetivos.

Considereei o produto muito complicado de utilizar

11 respostas

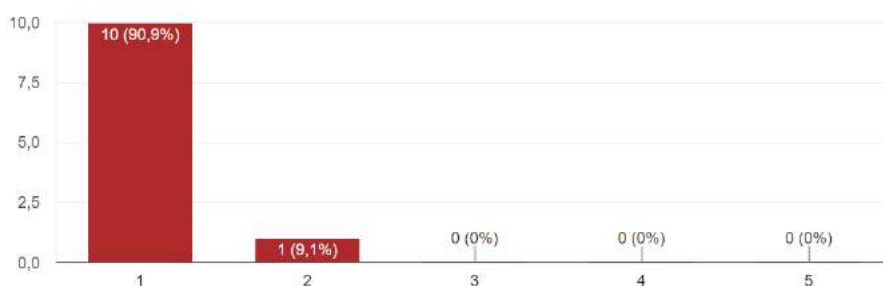


Figura 46 - Resultado SUS Pergunta 8

Pode-se concluir com o resultado obtido à pergunta 8, e tal como definido anteriormente neste documento, um dos principais objetivos da aplicação móvel foi cumprido com sucesso, sendo este o facto de ser simples de utilizar por qualquer pessoa.

5.2 Testes de Experiência do Utilizador e Validação de Requisitos

Os testes que se seguem contaram com a participação de 11 inquiridos, dos quais, 72,7% encontram-se na faixa etária compreendida entre 24 e 30 anos; 18,2% entre os 19 e 23 anos e 9,1% entre os 31 e 40 anos. 45,5% dos participantes possuem uma Licenciatura; 27,3% o Ensino Secundário e 27,3% o Mestrado. Estes testes têm como propósito validar os requisitos definidos à aplicação móvel já descritos neste documento.

Cenários

Juntamente com o envio do questionário, fizeram-se as seguintes propostas de cenários:

1. Realizar uma marcação de check-up a um dos seus veículos;
2. Verificar se recebeu notificação e e-mail a alertar que o seu veículo já se encontra pronto;
3. Verificar se o serviço ao seu veículo foi a troca dos 4 pneus e se o valor total é de 442.80 €;
4. Verificar se têm algum lembrete para o mês corrente;
5. Verificar se têm pelo menos 1 veículo listado na aplicação.

Perguntas e Resultados

Nesta segunda fase do questionário pediu-se aos inquiridos que realizassem os seguintes testes à aplicação móvel descritos na tabela abaixo:

Tabela 7 - Testes Aplicação Móvel

Pergunta nº 1	Conseguiu instalar a aplicação?
Pergunta nº 2	Conseguiu registar-se na aplicação?

Pergunta nº 3	Recebeu um e-mail ao efetuar o registo?
Pergunta nº 4	Conseguiu efetuar o login na aplicação?
Pergunta nº 5	Conseguiu visualizar os seus automóveis?
Pergunta nº 6	Conseguiu efetuar uma marcação para realizar serviços ao seu automóvel?
Pergunta nº 7	Recebeu um e-mail com a confirmação da marcação dos serviços a realizar ao seu automóvel?
Pergunta nº 8	Conseguiu visualizar as marcações do seu automóvel?
Pergunta nº 9	Conseguiu visualizar os serviços efetuados ao seu automóvel?
Pergunta nº 10	Conseguiu visualizar lembretes (avisos)?
Pergunta nº 11	Recebeu alguma notificação? Exemplo: Estado do automóvel, inspeção, revisão, etc.
Pergunta nº 12	Conseguiu efetuar o logout da aplicação?

Verificou-se que os resultados finais foram todos de resposta “Sim”, uma vez que, todos os inquiridos conseguiram realizar os testes acima referidos com sucesso.

Sugestões e comentários colocados pelos participantes à aplicação móvel

No final deste questionário colocou-se uma pergunta aberta a pedir o feedback aos inquiridos. Na tabela seguinte pode-se verificar algumas sugestões e comentários:

Tabela 8 - Sugestões e comentários dos participantes do questionário

1	<i>“Achei esta aplicação extremamente útil por nos dar uma noção mais precisa do estado do nosso carro sem que tenhamos de estar constantemente preocupados. A app é também bastante intuitiva e “user friendly”.</i> ”
2	<i>“No momento de Registo, acho que a alocação da password deveria ter dois campos. Ou isso, ou poder colocar visível para saber se o que escrevi na pass está OK. Para depois não acontecer de ter colocado mal e ter que andar a pedir logo nova pass e ainda nem entrei no sistema.”</i>
3	<i>“O icon do logout devia ter algum texto a indicar a função deste.”</i>
4	<i>“Grafismo mais elaborado na secção dos veículos. Data e hora das</i>

	<i>marcações muito complexas.”</i>
5	<i>“Simples, rápida e fácil de usar.”</i>
6	<i>“Caso me tenha escapado por não ter visto bem, talvez falte um "home button", pois após fazer a marcação tive que retroceder várias vezes até chegar à página mãe.”</i>
7	<i>“Aplicação bastante simples e intuitiva”</i>
8	<i>“Gostei bastante. Útil para lembrar o estado do carro”</i>
9	<i>“Adorei a aplicação.”</i>

Pode-se assim verificar que de forma geral os inquiridos gostaram da aplicação móvel.

5.3 Conclusão

Em jeito de conclusão a este capítulo de testes e análise de resultados, pode-se concluir que ambas as aplicações apresentam bons resultados em termos de usabilidade. Verificou-se, também, que os testes à aplicação móvel correram todos com sucesso, o que demonstra que a aplicação e *API* funcionaram e estiveram em constante comunicação sem qualquer problema. Com o feedback das pessoas inquiridas pode-se aferir que, apesar de este sistema ser útil, ainda precisa de ser trabalhado e melhorado para que possa responder positivamente ao mercado.

6. CONCLUSÕES E PERSPETIVAS DE DESENVOLVIMENTO FUTURO

O presente documento teve como objetivo apresentar o desenvolvimento de uma plataforma de gestão, website e aplicação móvel para oficinas de automóveis. O trabalho dividiu-se em duas fases: A primeira foi dedicada ao estudo de soluções do mesmo ramo existentes no mercado, assim como ao estudo de soluções tecnológicas essenciais para proceder ao desenvolvimento da aplicação, através de uma pesquisa de alguns softwares semelhantes de gestão de oficinas.

Também foram levantados os requisitos da aplicação assim como definida a arquitetura da mesma.

A segunda fase foi dedicada ao desenvolvimento das componentes assim como à fase de testes sobre as mesmas e correções de erros detetados.

A passagem pelas várias fases da dissertação permite no final chegar ao produto desejado. Através do estudo do Estado da Arte, feito no início da dissertação, foi possível analisar soluções existentes no mercado para a criação de plataformas web e aplicações móveis para oficinas de automóveis.

Foram analisadas algumas soluções consideradas como sendo as mais populares encontradas na literatura, bem como algumas soluções consideradas relevantes tendo em conta o produto final a desenvolver. Para além disso, foram analisadas e comparadas várias soluções tecnológicas que poderiam ser utilizadas nas aplicações desenvolvidas.

A análise dos requisitos permitiu definir com facilidade as funcionalidades a serem implementadas nas aplicações desenvolvidas. Procedeu-se à definição da arquitetura do sistema, identificando elementos relevantes para a mesma, como por exemplo, a existência de servidores que farão a ligação com o alojamento da base de dados e respetivos ficheiros web da plataforma, assumindo uma importância crescente na busca e disponibilização de informação oficina vs cliente.

Com a conclusão deste trabalho considera-se importante desenvolver num futuro as seguintes funcionalidades:

- Na aplicação web, melhorar a plataforma de gestão para o utilizador, permitindo que este possa inserir os seus veículos; possa realizar marcações mais detalhadas, facilitar a visualização do histórico do automóvel e que tenha mais informações e infografia pertinentes no seu painel.
- Na aplicação web, o administrador antes de realizar um serviço ao veículo, possa fazer a análise e o levantamento de todos os danos do automóvel e permitir a inserção desses dados.
- Na aplicação móvel, permitir que cada automóvel do utilizador tenha uma fotografia associada.

REFERÊNCIAS

- ABED, R. (2018, Setembro 12). *HYBRID VS NATIVE MOBILE APPS*. Obtido de http://www.ymedialabs.com/wp-content/uploads/2015/08/ymedialabs_native_vs_hybrid_V3.pdf
- Anant, K., & Chandran, P. (2018, Setembro 12). *Native Vs Hybrid Apps*. Obtido de <https://www.ijcter.com/papers/volume-2/issue-6/native-apps-vs-hybrid-appsionic-titanium.pdf>
- ARTSOFT. (2018, Outubro 1). Website Oficial ARTSOFT. Obtido 1 de Outubro de 2018, de <https://www.artsoft.pt/gestao-de-oficinas>
- Bernazzani, S. (2018, Novembro 16). What's the System Usability Scale (SUS) & How Can You Use It? Obtido de <https://blog.hubspot.com/service/system-usability-scale-sus>
- Bit Magazine, B. M. (2016, Janeiro 21). Aplicações móveis vão chegar em força aos serviços de apoio ao cliente. Obtido 1 de Dezembro de 2018, de Bit Magazine website: <https://www.bit.pt/aplicacoes-moveis-vao-chegar-em-forca-aos-servicos-de-apoio-ao-cliente/>
- Brooke, J. (1995, Novembro). *SUS - A quick and dirty usability scale*. Obtido de <https://hell.meiert.org/core/pdf/sus.pdf>
- Brooke, J. (2013, Fevereiro). SUS: A Retrospective. *Journal of Usability Studies*. Obtido de https://uxpa.org/sites/default/files/JUS_Brooke_February_2013_0.pdf
- CloudLinux. (2019). Obtido de <https://www.cloudlinux.com/>
- Django. (2018, Janeiro 30). Website Oficial do Django. Obtido 30 de Janeiro de 2018, de <https://www.djangoproject.com/>
- Doyle, B., & Lopes, C. (2005, Junho). *Survey of Technologies for Web Application Development*. Obtido de <https://pdfs.semanticscholar.org/1059/f3ee1e8f919729864253cad7db3191ef31fa.pdf>
- Express. (2018, Janeiro 30). Website Oficial Express. Obtido de <https://expressjs.com/>
- Heitotter, H., Hanschke, S., & Majchrzak, T. (2013). *Evaluating Cross-Platform Development Approaches for Mobile Applications*. Obtido de <https://www3.nd.edu/~cpoellab/teaching/cse40814/crossplatform.pdf>
- IBM. (2012). Native, Web or Hybrid Mobile-App Development - Whitepaper. Obtido 1 de Dezembro de 2018, de Computerworld website: <https://www.computerworld.com.au/whitepaper/371126/native-web-or-hybrid-mobile-app-development/>

- Ionic. (2018, Dezembro 1). Ionic Framework. Obtido 1 de Dezembro de 2018, de Ionic Framework website: <https://ionicframework.com/docs/intro/installation/>
- Kappel, M. (2018, Dezembro 12). 5 Ways to Build Killer Relationships With Customers. Obtido de <https://www.entrepreneur.com/article/282364>
- Laravel. (2018, Janeiro 30). Website Oficial da Framework Laravel. Obtido 30 de Janeiro de 2018, de <https://laravel.com/>
- Magic Web Solutions. (2019, Janeiro 30). The benefits of web-based applications. Obtido de <https://www.magicwebsolutions.co.uk/blog/the-benefits-of-web-based-applications.htm>
- Majeed, A., & Rauf, I. (2018, Setembro 26). *MVC Architecture: A Detailed Insight to the Modern Web Applications Development*. Obtido de <https://crimsonpublishers.com/prsp/pdf/PRSP.000505.pdf>
- Martins, A., Rosa, A., Queirós, A., Silva, A., & Rocha, N. (2015). *European Portuguese validation of the System Usability Scale (SUS)*. Obtido de <https://www.sciencedirect.com/science/article/pii/S1877050915031191>
- MySQL. (2018, Dezembro 1). Site Oficial MySQL. Obtido 1 de Dezembro de 2018, de <https://www.mysql.com/>
- Officegest. (2018, Outubro 1). Software para Oficinas | Officegest Software de Gestão Online Portugal. Obtido 1 de Outubro de 2018, de <https://officegest.pt/software-para-oficinas/>
- Phoenix. (2018, Janeiro 30). Website Oficial Phoenix. Obtido de <https://hexdocs.pm/phoenix/overview.html>
- Pop, D.-P., & Altar, A. (2014). Designing an MVC Model for Rapid Web Application Development. *Procedia Engineering*, 69, 1172–1179. <https://doi.org/10.1016/j.proeng.2014.03.106>
- Red & Yellow. (2019, Fevereiro 2). *Web development and design*. Obtido de <https://www.redandyellow.co.za/content/uploads/2017/11/eMarketing-Chapter-6-Web-development-and-design.pdf>
- Ruby on Rails. (2018, Janeiro 30). Website Oficial do Ruby on Rails. Obtido 30 de Janeiro de 2018, de Website Oficial do Ruby on Rails
- Saleh, H. (2017, Maio 31). MVVM architecture, ViewModel and LiveData (Part 1). Obtido de <https://proandroiddev.com/mvvm-architecture-viewmodel-and-livedata-part-1-604f50cda1>
- Spring. (2018, Janeiro 30). Website Oficial do Spring. Obtido 30 de Janeiro de 2018, de <https://spring.io/>
- Statistica. (2019, Janeiro 3). Number of free mobile app downloads worldwide from 2012 to 2017 (in billions). Obtido de <https://www.statista.com/statistics/241587/number-of-free-mobile-app-downloads-worldwide/>
- Tarhini, A. (2011, Janeiro 22). Concepts of Three-Tier Architecture.

- Usability.gov. (2013, Junho 5). Website Usability.gov. Obtido de <https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html>
- Xanthopoulos, S., & Xinogalos, S. (2013). A Comparative Analysis of Cross-platform Development Approaches for Mobile Applications. *Proceedings of the 6th Balkan Conference in Informatics*, 213–220. <https://doi.org/10.1145/2490257.2490292>
- XD Software. (2018, Outubro 1). Site Oficial XD Software. Obtido 1 de Outubro de 2018, de <https://www.xdsoftware.pt/solucoes-xd/gestao-comercial/xd-auto>
- Yii2. (2018, Janeiro 30). Website Oficial Yii2. Obtido de <https://www.yiiframework.com/doc/guide/2.0/en/structure-overview>

ANEXOS

Anexo I – Capturas de Ecrã Website

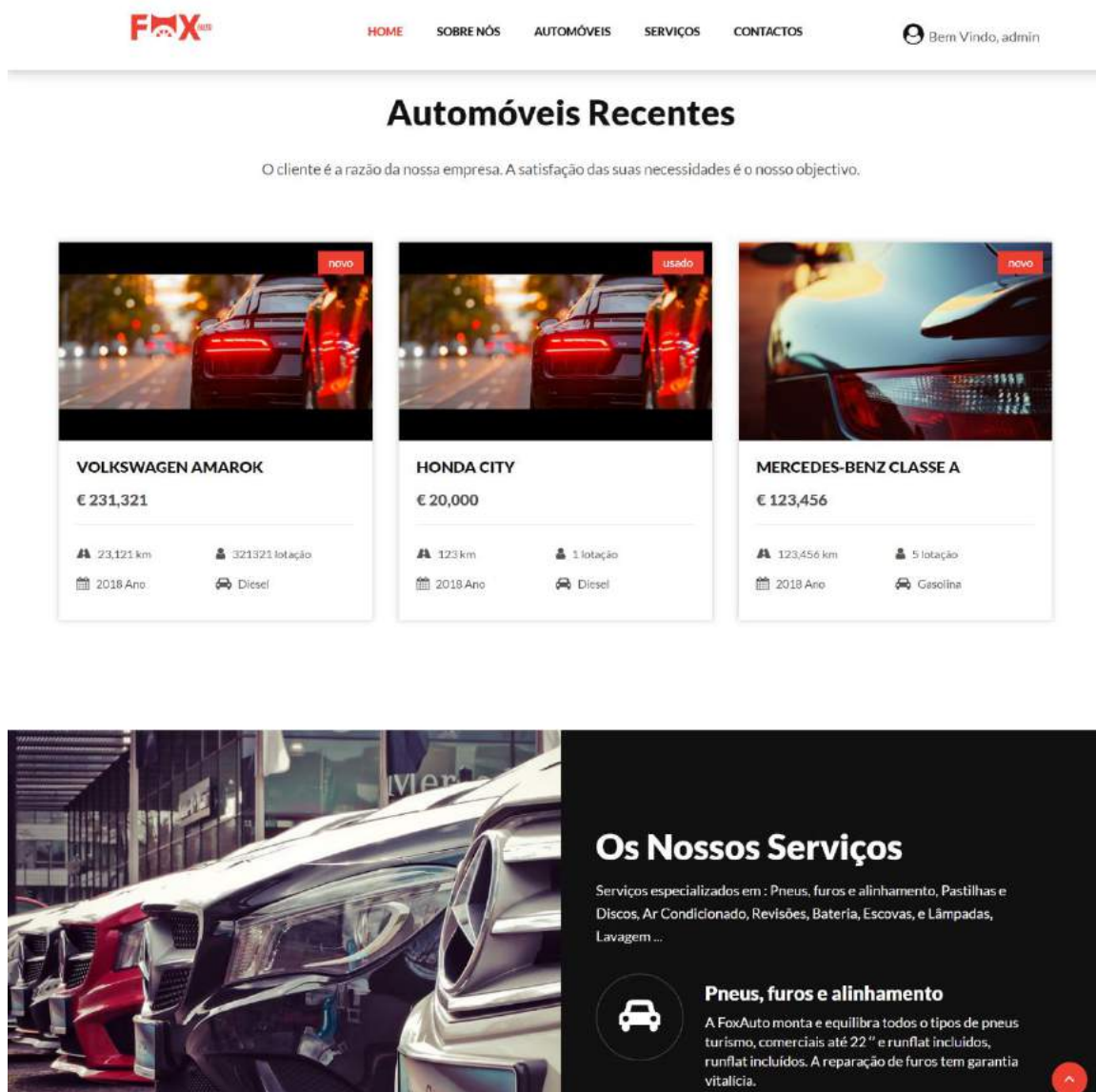


Figura 47 - Plataforma Web: Home



Figura 48 - Plataforma Web: Home Continuação



Figura 49 - Plataforma Web: Sobre Nós

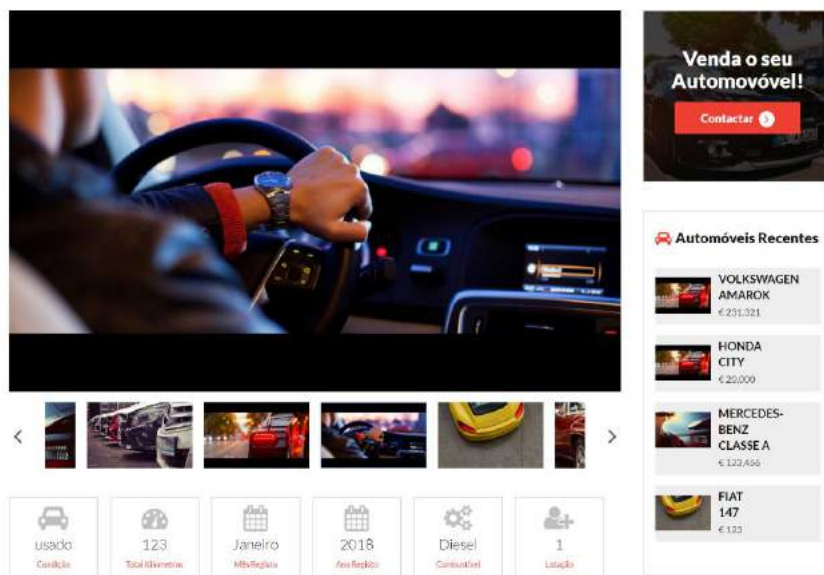
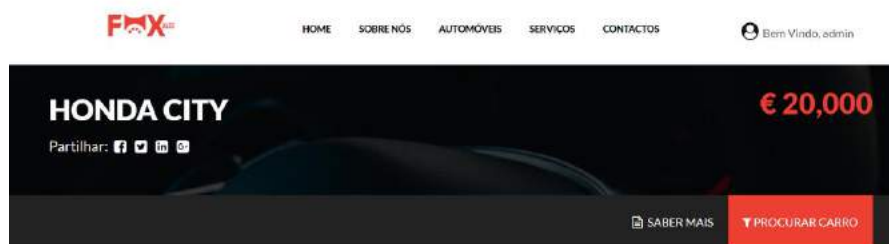


Figura 50 - Plataforma Web: Detalhes Automóvel

Especificação técnica		
DETALHES VEÍCULO		Observações
Marca	HONDA	
Modelo	CITY	
Versão	88	
Combustível	Diesel	
Mês Registo	Janeiro	
Ano Registo	2018	
Quilómetros	123	
Condição	usado	
Potência	213321	
Cilindrada (cm³)	213321	
Vin	213321	
Segmento	Utilitário	
Cor	Beje	
Metallizado	Não	
Lotação	1	

Figura 51 - Plataforma Web: Detalhes Automóvel Continuação

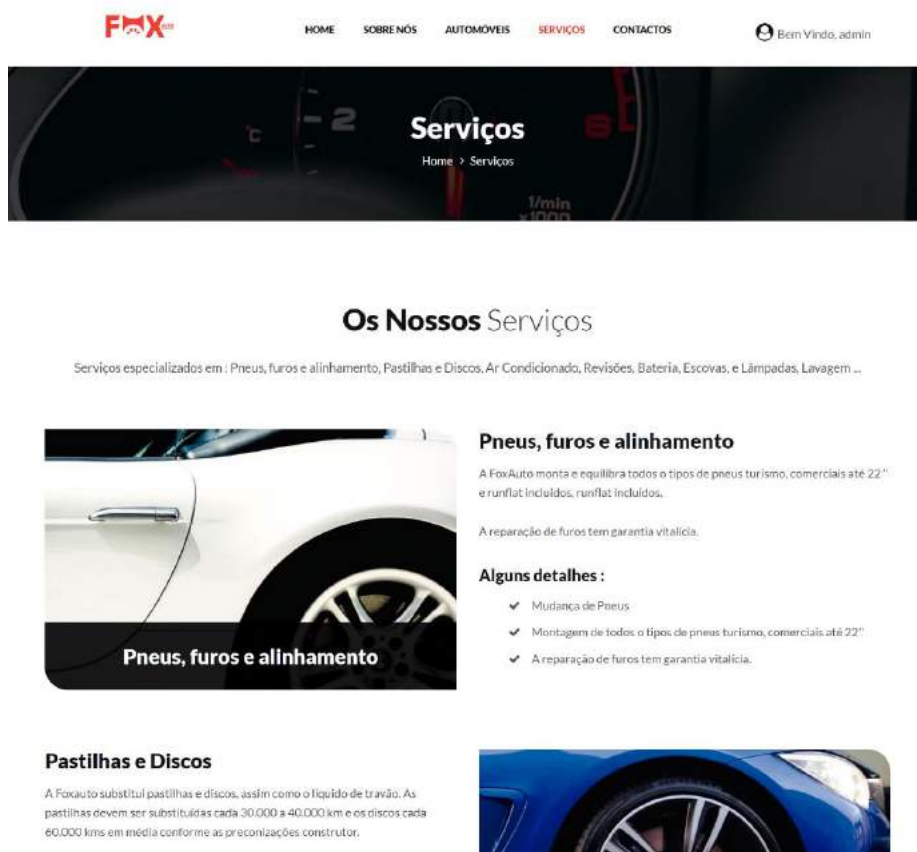


Figura 52 - Plataforma Web: Serviços

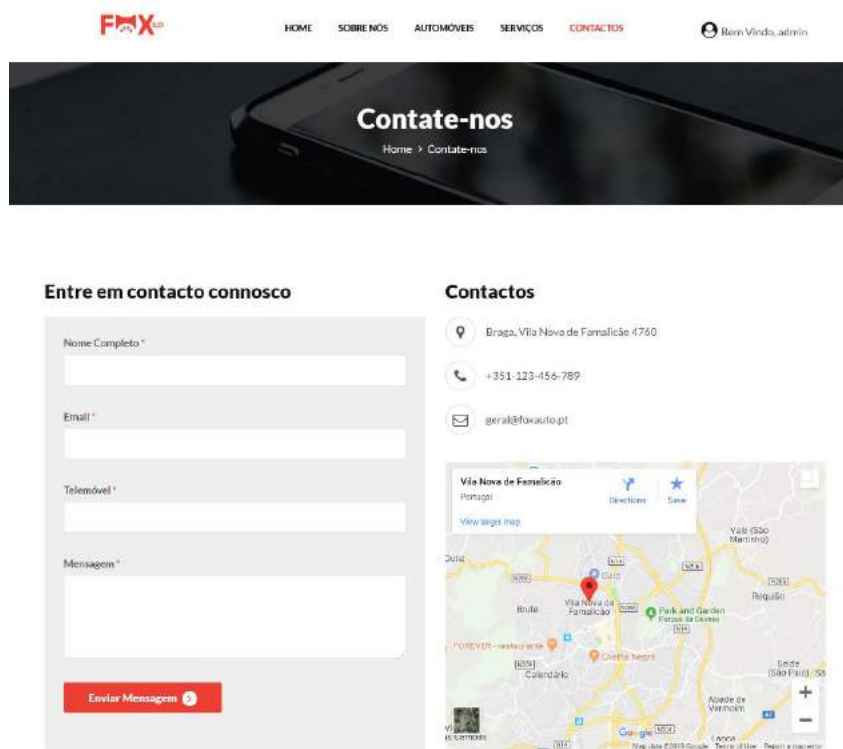


Figura 53 - Plataforma Web: Contactos

Anexo 2 – Capturas de Ecrã Plataforma de Gestão

FoxAuto Bem Vindo, admin

Adicionar Novo Cliente

Detalhes

Detalhes Cliente

Primeiro Nome *
Primeiro Nome

Morada
Morada

Último Nome
Último Nome

NIF
NIF

Data de Nascimento
Ano-Mês-Dia

Localidade
Localidade

CC / BI
CC / BI

Código Postal
0000-000

Contacto
Contacto

Status
Ativo

Detalhes Login

Email *
Email

Username *
Username

Password *
Password

Confirmar Password *
Confirmar Password

* Campos Obrigatórios

Criar Cliente

Figura 54 - Plataforma Web: Adicionar Novo Cliente

FoxAuto Bem Vindo, admin

Administradores

admin

Detalhes **Editar**

Informações de Contacto

Email: admin@gmail.com

Telefone: 911111111

Informações Cargo/Status

Cargo: admin

Nível: master

Status: ativo

Informações Adicionais

Aniversário: 2018-04-07

Morada: Rua do Casa

Localidade:










NIF: 123456789

Código Postal:

CC / BI:

Data Registo: 2016-11-23

Carros

Matrícula	Marca	Modelo	Data Matrícula	Ação
AA-00-AA	BMW	330	2018-04-07	  
AA-11-AA	ADURA	INTEGRA	2018-04-07	  
AA-22-AA	ASTON MARTIN	V8 VANTAGE COUPE	2018-04-08	  

Serviços







Número Série	Data Registo	Estado	Ação
SV 2018 16417616	2018-04-16 17:52:01	Em pagamento	  
SV 2018 144183918	2018-04-14 18:39:43	Liquidado	  

Figura 55 - Plataforma Web: Detalhes de Cliente

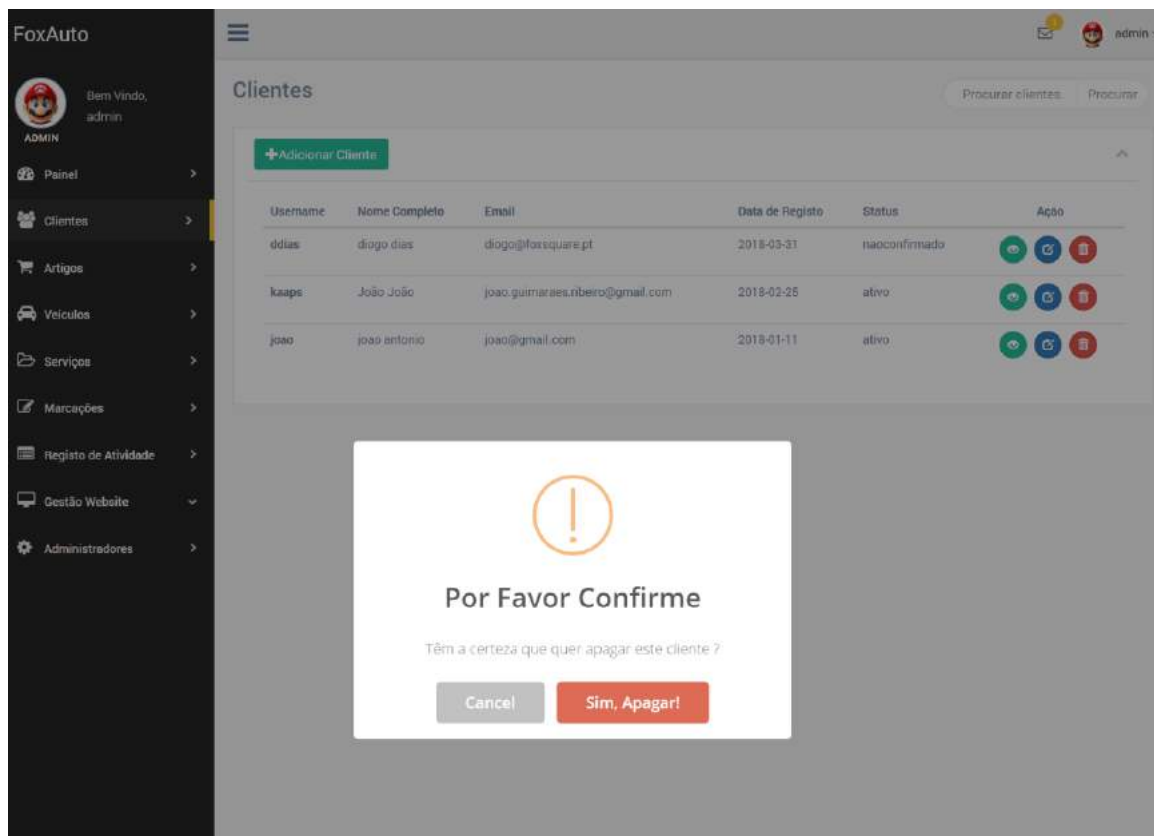


Figura 56 - Plataforma Web: Apagar Cliente

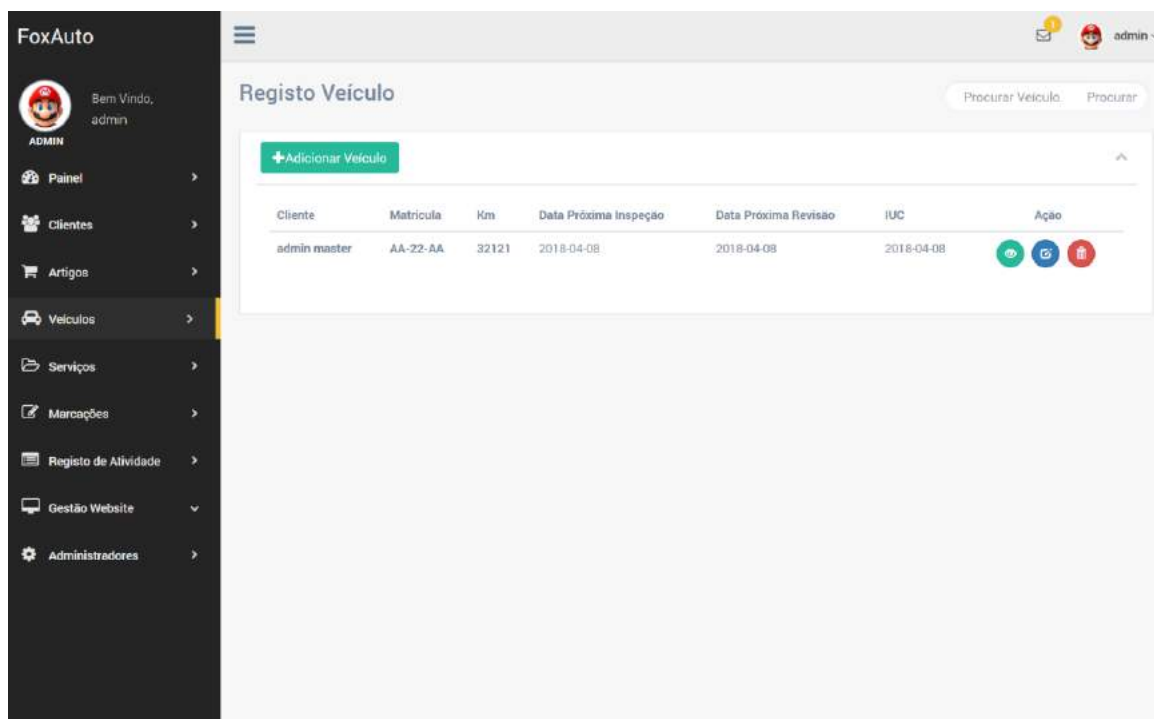


Figura 57 - Plataforma Web: Registo Veículo

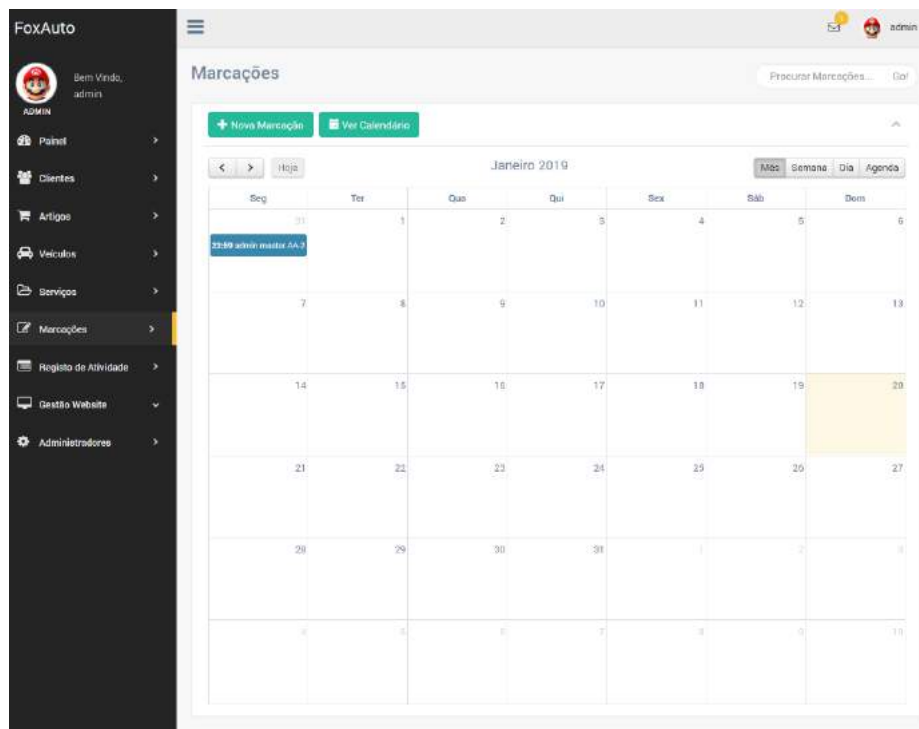


Figura 58- Plataforma Web: Marções (Calendário)

Figura 59 - Plataforma Web: Adicionar Novo Automóvel para Venda